February 1994

Report No. STAN-CS-TR-94-1503

Also numbered KSL-93-71

Thesis

DECIDING WHETHER TO PLAN TO REACT

by

Vlad G. Dabija

Department of Computer Science

Stanford University

Stanford, California 94305

DECIDING WHETHER TO PLAN TO REACT

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Vlad Grigore Dabija

December 1993

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

B Hayes-Roth

Barbara Hayes-Roth
(Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jean-Claude Latombe
(Co-Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

David M. Gaba

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Oussama

Oussama Khatib

Abstract

Intelligent agents that operate in real-world real-time environments have limited resources. An agent must take these limitations into account when deciding which of two control modes - planning versus reaction - should control its behavior in a given situation. The main goal of this thesis is to develop a framework that allows a resource-bounded agent to decide at planning time which control mode to adopt for anticipated possible run-time contingencies. Using our framework, the agent: (a) analyzes a complete (conditional) plan for achieving a particular goal; (b) decides which of the anticipated contingencies require and allow for preparation of reactive responses at planning time; and (c) enhances the plan with prepared reactions for critical contingencies, while maintaining the size of the plan, the planning and response times, and the use of all other critical resources of the agent within task-specific limits. For a given contingency, the decision to plan or react is based on the characteristics of the contingency, the associated reactive response, and the situation itself. Contingencies that may occur in the same situation compete for reactive response preparation because of the agent's limited resources. The thesis also proposes a knowledge representation formalism to facilitate the acquisition and maintenance of knowledge involved in this decision process. We also show how the proposed framework can be adapted for the problem of deciding, for a given contingency, whether to prepare a special branch in the conditional plan under development or to leave the contingency for opportunistic treatment at execution time. We make a theoretical analysis of the properties of our framework and then demonstrate them experimentally. We also show experimentally that this framework can simulate several different styles of human reactive behaviors described in the literature and, therefore, can be useful as a basis for describing and contrasting such behaviors. Finally we demonstrate that the

framework can be applied in a challenging real domain. That is: (a) the knowledge and data needed for the decision making within our framework exist and can be acquired from experts, and (b) the behavior of an agent that uses our framework improves according to response time, reliability and resource utilization criteria.

Acknowledgements

I would like to thank my adviser Barbara Hayes-Roth for her significant support and guidance throughout my three years here at Stanford. Through many long discussions, Barbara helped me clarify my thoughts. She has always challenged my ideas in constructive ways and often provided alternative suggestions and views from different perspectives. Most significantly, she always challenged me to look for more general solutions or more meaningful results in my research. She taught me important lessons on how to validate my work and how to present it. I have always appreciated Barbara's availability when I needed her advice most; on so many occasions she found the time to see and advise me with very little or no notice at all. Last but not least, she suggested lots of tips on how I can give my daughter a better education.

I would like to thank my co-adviser Jean-Claude Latombe for his advice, both while supervising my research and during the time I was his teaching assistant. His thought-provoking questions and perspective were very helpful in enlarging the scope of my research. I am indebted to Jean-Claude for the significant time pressure he exerted on me to complete my thesis during 1993, without which it could have taken me much more time to complete this research.

I was very lucky to benefit from the advice and medical expertise of Dave Gaba during my work here. I thank him for his patience and the time he took to supply me with expert data and to give me his feedback on my experimental results. In numerous occasions he commented extensively on my

ideas from his perspective both as an expert in anesthesiology and an expert pilot. His computer science oriented comments were very helpful in crystallizing many ideas of my work.

I would like to thank Oussama Khatib for his advice on my work. He has pointed out related problems to my initial ideas, and challenged me to work out solutions to them.

Serdar Uckun has provided me repeatedly with expert data in both medical and car driving domains. I have also benefited from numerous discussions with him regarding my thesis or parts of it as well as other related topics. I want to thank him for all his time and efforts.

I have had innumerable discussions with my office mate David Ash on a huge variety of subjects, related or not to this thesis. He was always a step ahead of me in completing his Ph.D. program, and has always been a tough act to follow. The fact that we had our oral examinations in the same day was probably no coincidence. He has also provided me with a detailed description of his reactive planner model, which I have used in some of my experiments.

I would like to thank to all the experts who have helped me with their expertise in the domains in which I have conducted my experiments: David Gaba, Serdar Uckun, David Ash, Alex Brousilovsky, Lee Brownston, Janet Murdock, Rich Washington and Michael Wolverton. They have not only provided the domain data needed for the framework, but they have also evaluated the system's performance.

My work has also benefited from numerous discussions with: Lee Brownston, John Drakopoulos, Mike Hollander, Philippe Lalanda, Jan Eric Larsson, Alex Macalalad, Philippe Morignot, Janet Murdock, Marcel Schoppers, Evangelos Simoudis, Henny Sipma, Rich Washington, Michael Wolverton. They have all contributed on many occasions their comments and advice.

I would like to thank Ed Feigenbaum, Rich Fikes and Bob Engelmore for supporting my work at the Knowledge Systems Laboratory. My thanks also go towards the administrative team of KSL: Grace Smith, Peche Turner and Michelle Perrie, for a wonderful job which has made any bureaucracy related things unknown to me.

Two other groups of people have given me wonderful opportunities during my years at Stanford, and I want to thank them here. Dr. Shogo Nishida and Dr. Katsuhiko Tsujino gave me the opportunity to spend three wonderful months in their research group at the Mitsubishi Central Research Laboratory in Japan. Largely due to them, my work there was extremely productive, and my touristic experience was unforgettable. My visit there was the beginning of a wonderful friendship with them. Prof. G. Schweitzer and Nadine Tschichold-Gürman gave me a similar opportunity in Switzerland, and I would like to thank them as well.

I would also like to thank Ranan Banerji and Val Breazu-Tannen for their support and encouragement, as well as their unlimited confidence in my abilities. Their support made this entire endeavor possible.

Financial support for my work was provided by ARPA Grant NASA NAG 2-581: Intelligent Systems for Science and Engineering Applications.

Finally, I would like to thank my parents, Antoaneta and Constantin, for their love, sacrifice and support in everything I have ever done, and for encouraging me to pursue my studies. The pride they derived from my work has given me the strength to complete this thesis. I wish to thank my two years old daughter Dominique (born between the theory and the AI comprehensive exams) for being such a wonderful child and for the beautiful moments she gave me, which have recharged my batteries many times during my work on this thesis. Last but by no means least, I want to thank my best friend and wife Tatiana for her infinite patience during the long hours I put into this work. She gave me the opportunity to begin this work and then supported me all along the way. She provided me with unlimited love and support and the right balance of encouragement and restrain to keep me going through the most difficult moments. It is to these four wonderful people that I affectionately dedicate this dissertation.

To: Toni, Puiu, Tatiana and Dominique.

Table of Contents

List of Tables

List of Illustrations

Chapter 1
Introduction

How should an intelligent agent prepare to satisfy a goal, while being able to respond to the great variety of contingencies that might impede its achievement of goals? Short answer: through planning. For a more comprehensive answer, you may want to read this thesis. It may provide you with a partial answer to this question, but it may also raise many other questions.

Many AI research resources have already been devoted to finding solutions to the problem of planning, usually defined as choosing the next step or steps for the execution of a system, based on knowledge of the present situation, the system's goals, and the operators available. The essence of planning in AI is the ability to reason about actions and their effects, and equally important, this reasoning process can take place before the actual execution starts. Therefore, it must deal with all the uncertainties due to the fact that the actual situation at execution time can only be assumed at planning time, when many characteristics of the environment either cannot be taken into account, or simply cannot be known. Many activities in Computer Science can be regarded as instances of planning. One example is programming, which requires making decisions (at planning - i.e. programming - time) about actions to be performed later, at program execution time, based on expectations about the environment in which they will be executed. A computer program is a formal specification of how the resources of the computer will be applied to solve a given problem. Although

conventional plans are not synonymous with programs, as also argued in [Drummond, 1989], we briefly use the analogy here for explanatory purposes. The more complex and unpredictable the execution environment is, the more contingencies can occur during a program execution. The programmer must therefore prepare the computer to properly respond to as many of these contingencies as possible, while still keeping the program within the computer resources, that is, it must still be small enough to fit in memory and must still be fast enough to give an answer in a required amount of time. The same situation occurs in all other domains in which planning is required.

A special kind of planning is reactive planning, i.e. building a set of specific perception-action rules stored in a computationally efficient form [Brooks, 1986; Agre & Chapman, 1987]. From now on, we will call this type of planning *reaction*, as opposed to the conventional type of planning which we will call simply *planning* or sometimes, to clearly distinguish it from reaction, *conditional planning*. To continue our parallel with computer programming, interruptions, traps, exceptions, and error treatment routines in a program can be regarded as reactions. They are executed as response to a large number of specific situations, and are not necessarily intended to ensure the successful normal continuation of the program towards completing its final goal. Sometimes, they are just intended to allow the program to interact gracefully with the environment or to help the program recover from a critical point and allow the user to intervene to facilitate the continuation of the program, or maybe to start the execution of another program, or even to write another program (to *replan*).

All the characteristics discussed so far for computer programming apply to most domains where planning is needed as a means of ensuring proper behavior of the system, before starting the actual execution of that system to achieve a given goal. Such domains range from "high-level" cognitive, symbolic domains like medical fields (e.g. anesthesiology, intensive care monitoring), to "low-level" manipulation domains like robot manipulator control. The common characteristics of all such domains is that their planning tasks can be (at least conceptually) translated into computer programs, and therefore conform to our previous discussion.

The same planning problem can be of very different levels of difficulty, depending on the assumptions made about the environment in which the plan is to be executed. For a well structured, "well behaved" environment which will not present "surprises" to the executing agent, the planning problem is much easier than for a more natural environment. In the latter case, many contingencies are possible during plan execution. We will call a contingency any state of the world entered by the executing agent while following a plan, that should not have occurred as a result of executing the plan up to that point. Contingencies are the effect of interactions between the agent and the environment; they occur because of: (i) predictable actions of the environment, or (ii) the unpredictability of the environment, or (iii) the unpredictability of the execution subsystems of the agent. In the real world, the number and variety of contingencies that can occur during the execution of a plan is unlimited. An ideal planner should take care of all these contingencies and build a "universal" plan [Schoppers, 1987] for the agent. As has already been shown [Ginsberg, 1989], building such a plan is not feasible for interesting application domains, due to practical limitations of the agent's resources. However, many of these contingencies can be ignored, either because they do not seriously affect the execution of the plan or because they have an extremely low likelihood of occurrence. Some of the remaining contingencies may have a very high likelihood of occurrence while also requiring elaborate subplans to treat them. Therefore, these subplans should be included as conditional branches in the original plan. Other significantly less likely contingencies may allow for a very short time of response, while having disastrous consequences if the response does not occur in time. Such contingencies probably should be treated reactively. These reactions need not lead the agent to the final goal of the initial plan; it is enough if they can stabilize the situation, avoid the consequences of the contingency, and allow the planner to replan a comprehensive solution from the current situation to the final goal. Yet other contingencies, not extremely likely and without short term dramatic consequences, can be ignored at planning time and left for a possible replanning phase at execution time: when they appear, the agent (which is not under very high time pressure) can suspend execution and take its time to replan a solution from that situation to the final goal. This may involve either a complete solution or, more frequently, a patch to bring the agent back to one of the states in its original plan from which it can continue

execution (one such mechanism was implemented by the triangle tables used in STRIPS [Nilsson, 1984]).

From the above discussion we can derive the two basic control modes for an agent that must deal with such contingencies: planning and reaction. By planning we will understand here both building a course of action *before* starting its execution and dynamic replanning, i.e. interleaving planning with execution. Each of these two modes has its advantages in certain circumstances, and we shall summarize them here. [Hayes-Roth, 1993] presents a complete discussion of these characteristics.

Among the strengths of the planning model is the fact that plans can be built to have a set of desirable global properties regarding the goals to be attained and the resources of the agent. The side effects of the actions to be executed as part of the plan can be carefully taken into account and analyzed before execution begins. These properties are achieved by taking into account complete descriptions of the states of the world as they are predicted by the planner. Of course, these states will conform to reality only if the environment behaves according to the model that the planner has about it. The more incomplete this model is, the more uncertainty in the behavior of the environment, and the more uncertainty about the actual states that will be encountered by the agent during plan execution. The final plan has a high degree of coherence and is easily comprehensible by a human user (this last point is very important in domains where the entire credibility of the system depends on how much a user can understand from the reasoning of the system, such as medical domains). The plan generated by a conditional planner usually makes a close approximation of the optimal usage of the agent's resources. Finally, the planned actions can be executed promptly at run time (since the agent simply follows a completely specified plan, in which the next action is taken according to the plan, maybe after evaluating the results of some tests in the case of conditional plans). However, the planning model has its weaknesses with respect to the real world. The two main disadvantages are: (i) the high computational cost of planning (which makes it necessary to carefully consider which contingencies should be exhaustively treated in this way - otherwise the time to build the plan may become prohibitive); and (ii) the inflexibility of the planned behavior - the agent can only act in states of

the world which are specified in the plan, and its performance will degrade very abruptly with any variations to such states.

The reactive model constructs a set of goal-specific perception-action rules and stores them in a computationally efficient form. The main advantages of the reactive model are its flexibility of response to a larger set of run-time conditions (since each response is less carefully analyzed than in the previous case, and the response does not need to embody a complete solution to the final goal but can merely be an action to stabilize the situation and allow the time for replanning) and its short time of response (determined by the efficient way of storing the reactive plan). On the other hand, reaction still cannot anticipate, distinguish and store all runtime contingencies. It will therefore still exhibit precipitous failure in unanticipated conditions. But the main disadvantage of reaction is that it is taken after a superficial evaluation of the current state, and does not benefit from an in depth analysis of this state and the related action consequences. Therefore, while a reaction may be locally appropriate, its global effectiveness is uncertain.

The planning and reactive control modes are near the end-points of a theoretical continuum of control modes. Together with two other control modes (reflex and dead-reckoning), they form a two-dimensional space described in [Hayes-Roth, 1993]. Also analyzed there is the correspondence between the space of control modes and a two-dimensional space of control situations, as well as the effect of combining the control modes in different degrees on the quality of run-time behaviors in the corresponding space of control situations.

We believe that planning and reacting complement each other, and therefore we envision agents that: (a) plan courses of action designed to achieve goals under certain anticipated contingencies - conditional branches are built in the plan for the very likely contingencies that also require significant planning to reach the goal; (b) augment these plans with context-dependent reactions for noticing and responding to less likely, but important exogenous events; (c) control their behavior by following their plans, while simultaneously monitoring for and, when appropriate, executing reactions associated with particular phases of their plans; and (d) revise their plans when local reactions do not adequately address unanticipated events.

However, this complementarity of the planning and reaction control modes in intelligent agents is overlooked by many researchers today. Most planning research to date has been concentrated either towards just one of the two control modes, or when it attempts to combine them, the main purpose is to increase the reactive capabilities of the agent and to unload the conventional planner's responsibilities. In this latter case, the general assumption is that reaction comes for free, that is, either the agent's resources are unlimited or the reaction process does not use any significant amount of the agent's resources. Unfortunately, this is not the case in reality: any real agent has limited resources, and the reaction process may use significant amounts of the agent's resources. This fact has a couple of consequences: (i) a decrease in the reactive responsiveness of the agent (or equivalently an increase in its response time to a given contingency), which may make some reactions useless if they come too late, and (ii) a limitation in the number of reactions for which the agent can prepare in a given situation. This means that the agent must be more selective in the types of reaction it prepares for each situation, preparing the most important reactions and ignoring the others. In the following chapters we define and characterize the value of reactions and identify those characteristics of the agent and its working environment that influence the response capabilities of the agent to different situations that it may encounter in its working environment. Based on this analysis, we formulate a framework to decide, at planning time, which control mode to choose for contingencies that may appear during plan execution, that is, a framework to decide, at planning time, whether a certain situation requires special preparation for a possible reactive response, or whether it can be left for dynamic replanning at execution time. The problem is particularly important for planning the activity of an intelligent agent which must work in a dynamic, complex, unpredictable real-time environment.

The approach begins with a plan designed to achieve a goal and enhances it to cope reactively with critical contingencies, while maintaining the size of the plan and the planning and response times within reasonable limits. The framework can also be modified for the problem of deciding, for a given contingency, whether to prepare a special branch in the (conditional) plan or to leave the contingency for opportunistic treatment at execution time.

As an example, consider driving a car between two given locations. Before starting, the driving agent plans its route in some detail, including turns at intersections and expectations of achieving milestones along the way, in order to minimize travel time. It also prepares a conditional branch in its plan as an alternative route in case the original route is blocked at a certain intersection where blockage is highly probable. This conditional branch requires extensive planning resources but produces a complete solution that leads all the way to the final goal. Along the way, the agent in fact encounters unexpected heavy traffic and revises the remainder of its plan to take an alternate route. As it follows the revised plan, the agent passes a school, where it *watches carefully* for children who might suddenly run into the street. As it leaves the neighborhood of the school and enters an industrial area, the agent *forgets* about children and *watches for other contingencies* (e.g., railway crossings, trucks coming out of driveways). Note that the agent, while executing the plan, is prepared to react to certain contingencies at different stages of the plan, while using dynamic replanning to solve other contingencies.

Given certain conditions (like the time of day, the weather, the type of roads to be used) the agent prepares in advance for possible contingencies that may appear on certain portions of the trip. However, it does not include expectations for and responses to these contingencies as steps of the plan, since they are not essential for the plan to be executed successfully. On the other hand, if they happen and are not responded to properly, they may preclude the successful completion of the plan. Examples of such contingencies are: sliding on a slippery road in cold weather, an unsignalled object in the street during night time, a child running in front of the car from a nearby school, or a traffic jam at rush hours. Note that these contingencies were qualified by the characteristics of the situation in which they are likely to appear. For some such contingencies, a reactive response must already exist since the situation does not allow enough time for the agent to replan a solution. There exists an infinite set of such contingencies, so the agent cannot prepare to always react to all of them. Moreover, due to limited computational and non-computational resources, if the agent prepares for too large a set of contingencies in a situation, selecting the correct response for the one that actually occurs may become a too long process, thus rendering

the response ineffective. However, the responses to such contingencies do not need to include an entire solution to the main plan's ultimate goal; if the agent responds to them fast enough to avoid unwanted consequences, then it may take the time to replan the entire solution from there on. Since these contingencies are too many and not very likely, they do not warrant a complete conditional branch in the initial plan to lead to the final goal.

Therefore, we need a decision framework to guide the selection of contingencies for which a reactive response should be prepared at planning time. This need arises in many domains besides car driving (for example, in intensive care monitoring, anesthesiology [DeAnda & Gaba, 1991; Fish & al, 1991; Gaba & al, 1991; Gaba 1991], nuclear power plant operation [Woods & al., 1987]). Formulating this framework is an important step toward building the control engine of real-time intelligent agents with limited resources for such domains. The formulation and evaluation (theoretical and experimental) of such a framework is the topic of this research.

In the following chapter, we outline the problem in more precise terms. We define the notion of contingency and classify contingencies into types according to their importance and the way they should be treated by the agent (with conditional plans, with reactions, or simply ignored at planning time and left for dynamic replanning if necessary). We also characterize the domains where the framework developed here will be most applicable. Finally a review of related work points out similarities with other paradigms.

Chapter 3 presents the basic approach. After giving an intuitive solution for a simple problem in the driving domain and analyzing this solution, we present the details of the framework for the reaction preparation decision. We show how it can be used to establish the value of reacting to a contingency in a given situation and to make the decision of whether to plan to react to that contingency. The chapter closes with a discussion of how this framework may be modified and applied to decide whether a certain contingency, in a given situation, requires preparation of a complete branch in the initial conditional plan.

Chapter 4 discusses a proposal for a knowledge representation formalism for contingencies, reactions and situations, to facilitate the structuring of the planner's knowledge and its manipulation.

Chapter 5 presents a theoretical analysis of the framework presented in chapter 3 for deciding whether to plan to react to a given contingency in a given situation. A few formal properties are stated and justified, to support claims of generality and optimality (in terms of using the agent's resources) for the proposed formalism.

Experimental demonstrations are then presented and briefly analyzed in chapter 6. Three domains were used for this purpose: an everyday domain where everyone is an "expert" (car driving) and two highly specialized medical domains of expertise (anesthesiology and intensive care monitoring). Results include simulations of several models of human reactive behavior discussed in the literature. A demonstration in a complex, real-world application domain shows: (1) that the knowledge and data needed for the decision making process exists and can be acquired from experts in that domain; and (2) that the behavior of the agent improves (according to response time, reliability and resource use criteria) as a result of incorporating our decision framework in the agent's planning mechanisms.

After summarizing our work, we make in chapter 7 a few suggestions of natural continuations of this research, including applications of case based reasoning techniques for managing a library of reactive plans and a library of contingencies and reactions, and several applications of learning mechanisms to different parts of our framework.

Appendix 1 briefly presents the architecture of the reaction decision module and the interface for integrating the module in an intelligent agent.

Appendix 2 completes the vocabulary example started in chapter 4 for the driving domain. It presents a large enough grammar to represent most of the situations, contingencies and reactions used as examples from this domain throughout the thesis.

In appendix 3 we present the results of a number of experiments we have conducted in the anesthesiology domain, in order to provide further

evidence regarding the generality and applicability of our framework in real-world domains.

Finally, appendix 4 complements the presentation of intensive care monitoring domain experiments in chapter 6, by presenting a few complete sets of contingencies as they were ranked by our framework.

Chapter 2
The Problem

In this chapter, we outline the problem in more precise terms. We define the notion of contingency and classify contingencies into three types according to their importance and the way they should be treated by the agent (with conditional plans, with reactions, or simply ignored at planning time and left for dynamic replanning if necessary). We also give a characterization of the domains where the framework developed here will be best applicable and what its limitations are. Finally a review of related work points out similarities with other planning paradigms.

2.1. Contingencies

Let us consider first a more detailed version of the example presented in the previous chapter. Suppose the agent commutes each morning by car from home (starting point S) to the office (final goal G), as shown in figure 2.1. We will limit ourselves to the study of a small segment of the car's route between points A and E. Suppose the route comes to an intersection with a traffic light (point B). The fastest route between B and E is through C, which is the route the agent normally takes if the traffic light at point B is green. However, the driving agent knows that, if this traffic light is red, then many other traffic lights between B and E through C will be red when the car will reach them, thus making the journey very slow. In the same time, the agent knows that if,

at point B, it will take a right turn and go through point D, then it can reach point E (and therefore the goal G) much faster.

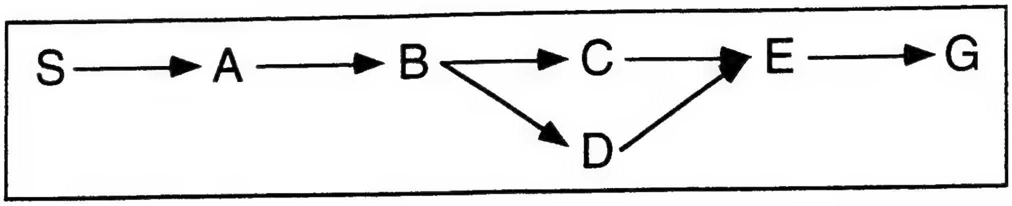


Figure 2.1. Conditional plan

The fact (and its associated state of the world) that the traffic light is red when the agent reaches the intersection at point B is a contingency, since it is not a result of the execution of the plan. In this case, the agent prepares a complete branch in the conditional plan to treat this contingency.

Suppose now that the point A in the plan built by the agent is a school in front of which the agent passes with its car. If the commute takes place at a time when children are at school, or go to school, the agent prepares to watch carefully for children who might suddenly run into the street. It also knows that in front of a school, a ball may suddenly pop up in front of the car. These and many other contingencies (some more of which will be considered in the demonstrations described later on) may appear during the time when the car is in the school zone A. As it leaves the neighborhood of the school and enters another area (e.g. an industrial area), the agent forgets about children and balls and prepares for other contingencies (e.g. railway crossings, trucks coming out of driveways, etc.).

Let us consider for a moment the following three contingencies which appeared in the previous example: the traffic light at point B, the child running into the street in front of the car, and the ball popping up in front of the car[1]. The common characteristic of these three contingencies is that they are not generated as a result of the execution of the plan. We define a *contingency* to be any state of the world entered by the executing agent while following a plan, which is not: (i) a direct consequence of executing the

1 In order to simplify the analysis for clarity of exposition, we have deliberately excluded the conventional driver's wisdom case that a ball popping up in the street is usually followed by a running child.

actions of the plan up to that point, or (ii) an exogenously generated state of the world assumed in the design of the plan. Therefore, a contingency does not necessarily affect the agent or the plan execution, and when a contingency does affect the plan, it is not necessary that it will negatively affect it. For example, a contingency may be a state which is not the current expected state according to the plan execution, but is a state which should have been reached along the way, after executing some more steps of the plan. The agent may detect it and use it to skip the unnecessary steps in the plan, for example in the same way as it was done with triangle tables in [Nilsson, 1984]. To simplify the exposition, from here on we will use the term *contingency* to also mean any fact or sign that was not expected as a result of the plan execution, and which may indicate that a state is a contingency according to the previous definition.

The three contingencies presented above are very different in nature, and will be treated differently by our agent. The traffic light contingency may happen very often (the actual probability to encounter a red signal is given by the length of time the signal is green divided by the length of time it takes the signal to complete an entire cycle, provided that the signal is not correlated with another signal previously encountered by the car and that the signal behaves independently of the amount of traffic that passes through it; for a two-way signal equally divided between the two directions of traffic, this probability is almost 0.5, though somewhat less because of the color yellow). Its likelihood of occurrence is significantly (one or more orders of magnitude) higher than that of the other two contingencies. The treatment of this contingency (by following an alternate route through point D to reach point E and then the goal G) also needs an elaborate plan which must be prepared in advance (otherwise, after turning right at the traffic light, the agent must stop and replan its route by possibly using maps, which may take a long enough time to wipe out any savings obtained by avoiding the traffic lights on the path through C). Therefore, the agent must prepare a conditional branch in the main plan for this contingency. This will use significant planning resources, but will have all the advantages associated with the planning control model discussed in the previous chapter.

The contingency defined by the child running in front of the car is much less likely to happen than encountering a red traffic light, even when driving in front of the school. This contingency has also a much higher

uncertainty about when and where it can occur. Thirdly, the plan to treat this contingency is much simpler (it is usually enough to brake and maybe to steer to the right, depending on the distance to the child); after taking the corrective action and avoiding the collision, the situation does not present any more dangers, so the agent can take its time to replan a course of action that will get it from the new state to the goal (this may be as simple as restarting the car, or as elaborate as finding an alternative means of transportation if the car was damaged by hitting a pole on the side of the road while avoiding the child). While the critical situation was avoided by a simple plan, the state obtained after its execution is unknown and may belong to a large set of very different states. Therefore, a comprehensive conditional plan to exhaustively treat all these states and preplan the agent's execution from them to the initial goal G may be prohibitive. The practical alternative is to treat such contingencies in a reactive manner, by attaching simple reactive plans to those points in the main plan where such contingencies may occur. After the reaction will yield a non-dangerous state for the agent, it can take its time to dynamically replan for a complete solution.

The third contingency stated before - the ball popping up in front of the car when driving along a school - is a little more probable than the child running in front of the car, but the likelihoods of the two contingencies are roughly of the same order of magnitude. However, in this third case, the consequences of hitting a ball with a car (especially with a relatively slow moving car in the vicinity of a known school) are significantly smaller than in the child case. Moreover, the side effects of making a dangerous maneuver to avoid the ball may outweigh by far the consequences of hitting the ball. Therefore, for such a contingency, the agent is much better off if it ignores it at planning time, thus conserving its limited resources for other more important contingencies.

To summarize the discussion in this section, we have identified there types of contingencies that may appear during the execution of a plan. They are classified according to the action taken by the agent at planning time to prepare for their occurrence at execution time. These types of contingencies are:

(i) contingencies for which the planner builds complete conditional branches, from the contingency state to the goal state, in the main plan;

(ii) contingencies for which the agent prepares reactive responses; they are combined into reactive plans by a reactive planner, and are attached to appropriate segments of the complete plan provided by the conditional planner;

(iii) contingencies ignored by the agent at planning time, either because their treatments can be left for dynamic replanning when they are encountered at execution time, or because they are considered less important than the contingencies included in the previous two categories, and the agent simply does not have the resources to prepare a reaction (much less a complete branch in the plan) for them.

The justification for this classification is mainly related to the limited resources that a real agent can use. For a few contingencies, the agent can generate complete plans and combine them in a conditional plan. However, the agent's limited planning and execution resources do not allow for too many contingencies to be treated this way. Still, the agent can prepare at planning time reactive responses for a larger set of contingencies; these responses will not ensure full solutions to the goal state, but they will give the agent the possibility to dynamically replan its actions at execution time. But in no case can a real agent with limited resources prepare for all possible contingencies in a real world application domain. Many of these contingencies must be ignored at planning time. The problem addressed in this thesis is how to decide which contingencies to select for preparation of reactive responses, and which to ignore at planning time.

2.2. Summary of the Problem

The example problem outlined in the previous section highlights many aspects of the general problem with which we are concerned. We shall define here this problem more precisely, and then we will propose a solution for it in the next chapter.

In all our previous discussion we have referred to *reaction planning* as a conscious form of preparing condition-action behavior. That is, the agent consciously prepares, before starting the actual execution, a set of perception-action rules for a certain segment of the plan. They are to be executed by high level execution mechanisms of the agent similar to those that execute the main plan, and are not intended for execution by a "lower level", higher priority execution mechanism which may be part of the agent architecture (like the one proposed by [Brooks, 1986; Kaelbling, 1987]). Actually, the agent will resort to a reaction to a contingency only if it has no conditional branch in the plan at that stage during the execution, and will consciously take the decision to try to use reaction in that situation. This does not mean that we specifically prohibit in our agent architectures any lower level execution mechanisms which have the ability to react faster and with higher priority to certain contingencies. It only means that we are not concerned with such precognitive types of reaction (e.g. locomotion type reactions like avoiding obstacles by a moving robot). We are only concerned here with contingencies to which such reaction mechanisms cannot respond. On the other hand, if the agent architecture does not include such low level reaction mechanisms, then the contingencies to be treated by them may join the set of contingencies which are analyzed by the higher level cognitive mechanisms of the agent using the framework proposed in this work.

Since we will talk more in the following section about the characteristics of the domains in which this work is best applicable, We will simply say here that we are particularly interested in planning the activity of an intelligent agent with limited resources and multiple goals working in a dynamic, unpredictable, real-time environment. The agent must itself act in real-time, i.e. be “predictably fast enough for use by the process being serviced” [Marsh & Greenwood, 1986]. In order to behave properly, the agent must plan its actions ahead of time, and then monitor the plan execution and be prepared to respond to contingencies that may appear during this execution. This emphasizes two orthogonal qualities that the agent must exhibit: sensitivity to run-time contingencies and commitment to specific goal-oriented actions. Such behavior can be accomplished by combining the two fundamental control modes mentioned before: planning and reacting.

As will be shown in section 2.4, most research to date is concerned either with employing only one of these control modes, or simply attempts to turn a system to become increasingly reactive and rely as little as possible on planning. These works concentrate mainly on how to prepare reactive responses and tend to use them in such a way as to substitute regular planning. Our approach differs from these others in its recognition of the complementary strengths and weaknesses of the two modes, and in its full integration of planning and reacting within a single agent.

Our premise is that, whenever time and other resources allow, a dynamically planned response is never worse (and usually better) in responding to a contingency than a reactive response[2] previously prepared for it. There are several reasons for this assumption: (i) the replanned response is generated at execution time when more information is available, as opposed to planning time, when the reaction is prepared; (ii) when replanning, an agent has time to analyze all the relevant information and to search for the best available solution by planning a complete solution path to the goal, while in order to react, the agent may have only a few alternatives (in the reactive plan) to choose from and only a few tests to decide on the response, which must therefore be taken based on incomplete information obtained from an incomplete analysis of the current situation; (iii) if time is so limited that it cannot even perform all these tests, the agent may have to take a more general action hoping to improve the situation at least temporarily and to buy more time to look for a better solution. The reason we need to use reaction is that the replanned solution may be found too late and therefore be of no more use at the time it can be taken. Thus, we assume that the importance of regular planning makes it irreplaceable (due to the vast diversity of situations in real-world environments), but the agent's real-time performance can be significantly improved by preparing reactive responses for a limited number of critical contingencies that may be foreseen to appear during execution of the plan already built to achieve the main goal.

---

2 Again we stress that, in this work we study conscious forms of reaction, prepared at planning time and consciously taken, as opposed to precognitive types of reaction (e.g. locomotion type reaction).

By not including enough contingencies for reactive treatment, the performance of the agent will be suboptimal. On the other hand, by including too many such contingencies, the reactive response time becomes too slow, thus degrading the system performance once again.

Unless otherwise stated, we assume that, given a contingency, the agent knows of an action (maybe a small sequence of elementary actions) which, if applied reactively, either solves the problem generated by the contingency, or at least postpones its deadline long enough to allow for replanning of the entire solution.

The main issue for us then is to enable the agent, for each phase of the main plan, to select the right set of contingencies for which to prepare reactions. That is, our *problem* is to specify a decision framework which:

- ○ *given*:
  - ● an intelligent agent with:
    - ✧ capabilities:
      - ✦ planning and dynamically replanning
      - ✦ monitoring
      - ✦ reactive behavior
    - ✧ constraints:
      - ✦ limited resources
      - ✦ real-time performance
  - ● a (possibly conditional) plan by which the agent can achieve its current goal
  - ● a set of contingencies known to possibly appear at certain times during the plan execution, each with:
    - ✧ reactive responses associated with them
    - ✧ known characteristics associated with each such contingency (e.g. gravity of consequences, time deadlines) and with their reactions (e.g. resource requirements)
- ○ *enable* the agent to decide at planning time on how to select *a "rational" subset* of these contingencies (according to a desired behavior pattern) for which the reactive responses should be attached to the main plan

(while preserving the real-time responsiveness of the agent to all these contingencies, given its limited resources).

We have used the word "rational" in the previous definition, and it needs some disambiguation. A behavior of the agent in a given situation is defined by the order in which the agent classifies the set of contingencies for that situation, according to the value of reacting to them. For the same situation and set of contingencies, there are different behaviors that the agent may exhibit. Some of these behaviors may either not be suitable for that situation, or may even be considered abnormal, hazardous or even pathological. But there is at least one such behavior which is considered appropriate or normal for that situation, by the experts in the domain. It is even possible that there are several different behaviors that may be considered appropriate in a given situation. Each behavior is appropriate according to a behavior model, and in the literature there have been defined a number of such reactive behavior types for domains in which critical and stressful situations are common and very dangerous like aircraft flying [FAA, 1991], nuclear power plant management [Woods & al., 1987] or anesthesia [Gaba & al., 1991]. In most of the thesis we will refer to what is considered to be the "normal" behavior by experts in each domain from which we draw our examples. However, in section 6.3, we will discuss some other types of behaviors and how they can be translated and simulated with our framework.

One problem related to the one we stated before is conditional planning. As discussed before, there are three courses of action that an agent can take to prepare a response to a possible contingency: plan a conditional branch, plan a reactive behavior, or ignore the contingency at planning time. Our analysis will focus on how to decide whether to prepare a reactive response to a contingency, but the general framework which will be developed for this purpose is also applicable (with certain modifications) to the problem of deciding whether to prepare an entire conditional branch in the main plan for a possible contingency. In section 3.5 we will briefly discuss what are the changes that must be made to our formalism so that it can also be used to decide which is the set of contingencies for which conditional branches should be planned. However, in the rest of the thesis, we will assume that the agent has already built the complete conditional plan, and is only trying to augment it

with reactive responses to as many contingencies as possible being limited by its finite resources.

The selection criteria which we are looking for are much more complex than any utility measures (e.g., [Minton, 1990]) proposed so far. For example, in our approach, some of the contingencies associated with a situation may appear in practice with a very low probability, but they may be very critical if they occur, and thus are worth preparing for reactively and are also worth being remembered. This is in contrast with most of the research to date, which is mainly concerned with improving the systems' performance by caching into reactive plans the responses to the most frequently occurring contingencies.

But before reviewing the previous research in this domain, let us attempt to characterize first the domains in which the problem stated here is significant and where our solution framework is applicable.

2.3. Application Domains

Much of the planning work to date has concentrated on applications in artificial domains. Such domains are well-structured and well-defined by the system designer, which usually means that the entire set of possible contingencies is known in advance, and that this set is of a manageable size. The main implication of this is that the resource limitations of the agent can be ignored (particularly at execution time) with respect to the size of the plan, whether the main control mode employed is conditional planning or reactive planning, that is, we can always assume that we have a powerful enough agent to be able to respond in time to any of the contingencies that it knows about. This is clearly an artificial assumption which drastically simplifies the planning problem and limits the applicability of the solutions proposed.

By contrast, we are interested here in applying the planning paradigms to real-world domains and to allow the agent to operate in real-world (albeit closed and limited for practical purposes) domains. The main characteristic of such a domain and the agents operating in them is *real-time* defined by [Marsh and Greenwood, 1986] as "predictably fast enough for use by the

process being serviced". This means that the agent must be guaranteed to respond, at execution time, in a prespecified time limit to any contingency for which it has prepared a response at planning time. However, if an agent with limited resources prepares to respond to too many contingencies in a certain situation, than it may not be able to guarantee a timely response to the most time-pressured of these contingencies: e.g. it make take too long for the agent to discriminate among the possible contingencies for which it is prepared to react, from the time it detects a contingency and until it has to take the corrective action. An example of an interesting domain for our framework is the car driving domain, which will be used for exemplification throughout most of the thesis. If a child appears in front of the car at small distance, there is very little time for the agent to discriminate among the contingencies for which it is prepared to react in that situation and to decide what kind of contingency this is and how to react to it. For an agent with limited computational resources it may be therefore better not to prepare to react in the same situation for a much less critical contingency like a ball coming in front of the car, or a sudden loss in the radio signal, and so on.

These observations are valid in real-life domains because another of their characteristics: they are *very large*, both in the number and variety of contingencies that may appear (which has been noticed a long time ago in [McCarthy, 1977] when describing the qualification problem), and in the variety of corrective actions that may apply. Each corrective action applicable to a certain contingency may be better suited in some situation than in another one. Therefore, we will always consider pairs contingency-situation associated with each situation in which that contingency may arise and in which that response is the best to this contingency. For well-structured (usually artificial) or very limited domains where the number of contingencies and responses is limited, the framework described in this thesis is not necessary, since it is conceptually possible to use a more powerful agent which can take care of all the contingencies in each situation.

As seen before, real-world environments are usually *unpredictable*, that is contingencies may occur at any time, or at least *uncertain* in that the effects of actions and the actual state of the world after the execution of a plan step cannot be foreseen with utmost precision. Such domains are also usually *dynamic* in the sense that the state of the world may change without the

participation of our agent, for example, as a result of actions of other - cooperative or antagonistic - independent agents working in the same environment (e.g. there are other agents driving cars on the same streets as our agent and their paths may intersect[3]). In real domains some contingencies tend to appear associated with certain plan steps or situations and the likelihood of their appearance may be different for different situations, while others can appear at any time with the same likelihood. For example, it is always possible for a child to run into the street, or for a meteor to fall into the street or for the car to fall to pieces, but it is impractical for the agent to be on the lookout for all of these possible events all the time. Real-world domains also present a *huge variety of situations.* In each situation different contingencies can happen, and the same contingency may be viewed differently in different situations. In certain situations, some contingencies are more likely or more important than others. If the agent has to drive the car on a mountain road in winter, it should expect bumps or damaged portions of the road, or slippery roads, instead of, say, traffic lights. The agent should prepare for yet another set of possible contingencies in the case of driving on freeways. Also, the most effective responses associated with a contingency which may appear in different situations may be situation dependent. The agent should therefore be able to selectively prepare itself for the most critical contingencies in each possible situation along a prepared plan.

We should also note that some of the contingencies associated with a situation may appear with a very low probability, but they may be very critical if they occur, and thus are worth preparing for. This is in contrast with most of the literature to date, since most authors are mainly concerned with improving their systems' performance by caching the most frequently used plans.

We also assume that short plans (a single action or a small sequence of actions), if applied reactively, are usually enough to either solve the problem generated by the contingency, or at least to postpone its deadline long enough to give the planner the time needed to dynamically replan the entire solution under the new circumstances.

---

3 Hopefully not at the same time...

Most real domains which have the features described above are usually characterized as high level, knowledge intensive domains. Examples of such domains are some medical domains (e.g. intensive care monitoring, anesthesia), nuclear power plant operation, aircraft flying, car driving and so on. These are contingency-intensive domains, in which many contingencies can appear and in which some of these contingencies are very time-critical and / or with very high consequences, even if they do not appear with very high frequency. Although these domains also require (some more than others) significant skill development (by skill we mean here automatic, low-level, unconscious reflexes to certain contingencies), their main characteristic is that the process of planning and responding to contingencies is knowledge-intensive and thus uses significant high-level cognitive resources of the agent. Our framework can be in principle applied to any domain, but its value and effectiveness can be questioned for very well structured, artificial domains (like the blocks world) and for low-level, skill intensive domains (or such tasks in higher-level domains), like locomotion tasks (e.g. reflex obstacle avoidance) or fine-motion robot manipulation tasks (e.g. the peg-in-the-hole insertion problem), in which the number and diversity of contingencies is limited and well-known in advance.

Even for such limited but real domains, we can argue that our framework can be applicable as long as the resources of the agent involved are not powerful enough to completely remove the uncertainty in the domain. An example of such a domain is robot motion planning. The main problem here is the uncertainty, at execution time, in the position and orientation of the parts and of the robot (e.g., a manipulator) in the workspace. A class of planning methods developed for this problem deal with such uncertainty in a second phase of planning; in the first phase, plan skeletons and local strategies are produced, using path planning methods which assume zero uncertainty (i.e. no contingencies) [Latombe & al., 1991]. Then different methods are used to deal with contingencies generated by the aforementioned uncertainties. For example, SPAR [Hutchinson and Kak, 1990] adds verification and local recovery plans to reduce uncertainty and to prepare for possible failures. Similarly to the reactions used in our framework, these local recovery plans are only single, special-purpose actions (which may be entered by the user) and are associated with uncertainty-reduction goals a

priori. An inductive learning technique is used by [Dufay and Latombe, 1984]: a trainer module generates patches to be inserted in the ground plan. These are local strategies refining the ground plan, similar to our reactive plans attached to the main plan (e.g. rotate a card to insert it into a slot). The system further provides for the graceful degradation of its performance by allowing for entering rules on line if everything else fails. However, the most common technique for dealing with uncertainty-generated contingencies in this domain is skeleton refining [Lozano-Perez, 1976; Taylor, 1976]. A skeleton plan (or assembly description) appropriate to the task at hand is retrieved as initial plan and then iteratively modified by inserting complements (e.g. sensor readings) during a feedback planning or plan checking phase. The modification of assembly strategies to fit particular geometric environments results in building conditional plans. Then strategies are examined for likely failures and the planner generates tests (monitoring actions) and inserts corrective actions (which are either conditional branches, or reactive plans - e.g. if the robot manipulator is on the verge of overturning a workpiece by pushing it with a peg, then retract the hand a little to stabilize the situation and then replan the action). If the plan contains many such reactions to too many contingencies for the same situation, the agent may become too slow to respond to some of the most time-critical of these contingencies. The solution is to use the framework developed here to choose among these contingencies. Further refinements of the plan-skeleton paradigm include symbolical computations of the effects of uncertainties [Brooks, 1982] to identify and treat the most significant ones by making inferences about uncertainties and using them in computations, as well as using formal program proving techniques to deal with these uncertainties [Pertin-Troccaz and Puget, 1987]. All this discussion shows that, even if the robot manipulator programming domain is not, as a whole, a high-level knowledge intensive domain (in the sense defined before), the formalism presented here can still be applied if the set of uncertainty-related contingencies becomes too large and if their treatment requires conscious actions (as opposed to just locomotive reflexes).

Besides the domain characteristics, the agent's capabilities are also important in this discussion. If we have an ideal agent with unlimited resources and unlimited speed of computation, then the entire formalism may become useless, even in the real-world, high-level domains presented before.

However, if we are again interested in the real world, then it is only natural to assume that the *agent has limited resources* and that the number of contingencies for which it has to prepare exceed both its conditional planning capabilities, and its real-time execution capabilities. In such cases, the domain exerts time pressure on the agent's limited resources. Therefore, the agent needs to be able to decide which contingencies to prepare treatment for and which to ignore at planning time. These are the types of agents and domains for which the framework developed here is useful.

2.4. Related Work

We make here a brief review of other work that is relevant to the problem of how to combine planning and reaction to achieve the best performance of the agent in a particular environment. The purpose of this section is to place our work in the global context of related research and to outline its original contributions.

Planning (describing a set of actions expected to allow the agent to achieve a given goal) has been a central problem in AI since its very beginnings [McCarthy, 1958]. The techniques proposed have evolved considerably, and so have the application domains. We classify these techniques into several classes, according to the ways they combine the two fundamental control modes described before: *conditional planning* (also called here *classical planning* or simply *planning*) and *reactive planning* (also simply called *reaction*). These classes are:

(i) purely conditional planning techniques

(ii) purely reactive techniques

(iii) static combinations of planning and reaction

(iv) techniques to shift from planning to reaction

(v) techniques to decide at execution time whether to (re)act or to continue the replanning process

(vi) techniques to decide at planning time which contingencies to prepare reactions for

A lot of early planning work has been conducted towards specifying robust techniques for conditional planning. The systems produced (e.g. STRIPS [Fikes and Nilsson, 1971], NOAH [Sacerdoti, 1975], MOLGEN [Stefik, 1981], TWEAK, [Chapman, 1987] to almost randomly name just a very tiny subset since an exhaustive summary would be well beyond the scope of this section) were able to solve increasingly complex problems. Although some of them had facilities for monitoring their plans execution and responding to some contingencies (e.g. PLANEX for STRIPS [Fikes & al., 1972]), these facilities were very limited and worked only in well-structured domains, based on the existence of a state matching the contingency in the original conditional plan. More flexibility and higher response speed was needed to build systems for real-world tasks.

The need for reactivity to the dynamic aspects of the environment was addressed by building systems which operate on a perception-action basis without relying on an abstract representation of the environment [Brooks, 1991]. Horizontal layer decomposed systems [Brooks, 1986; Kaelbling, 1987] included such reactions while still being able to pursue high-level goals, but their reactions were limited to the types of locomotive, low-level precognitive reactions which we described earlier and which do not make the object of our work.

Realizing full reactive behavior (reaction plan planning) has been proposed through universal plans [Schoppers, 1987] which are exhaustive conditional plans, and therefore are prohibitively expensive to produce for any reasonably complex domain [Ginsberg, 1989]. Situated Control Rules [Drummond, 1989] are used for situation-based plan indexing, to reduce the non-deterministic choice in the case of plan nets. They may be used as an incomplete alternative to universal plans, in those cases when there is not enough time to build the entire universal plan. An incomplete universal plan may not contain any answer to a problem, while missing situated control rules do not necessary preclude a solution (which may be found nondeterministically); they only ensure a solution when they are specified. This approach maximizes the use of planning time and takes into account

planning resource limitations, but without taking into account any execution time limitations of the agent.

Pengi [Agre and Chapman, 1987] is a purely reactive planning system which uses sensory input to index structures for possible subsequent actions. However, Pengi cannot completely represent most real situations due to their uncertainty and the limited information available about other agents and processes.

Due to the shortcomings of pure reactive systems, researchers have subsequently concentrated on integrating planning with high-level reaction. [Firby, 1987] uses Reactive Action Packages like stored reactive plans to integrate planning and reactive responses. However, reactive planning is used without time considerations, while we allow the agent to try to dynamically replan its course of action if there is enough time to do it, and only prepare to react to critical events. [Hendler & Agrawala, 1990] implement reactive planning systems on a guaranteed scheduling, real-time operating system using the Dynamic Reaction model: an agent performs an activity until either its goals lead it to select some new action, or some event in the world forces it to react, thus integrating planning and reaction in a complex environment. [Georgeff & Lanski, 1987; Georgeff, 1989] propose an architecture (the Procedural Reasoning System) that is both highly reactive and goal directed. They store (reactive) plans, called Knowledge Areas, in procedural form, supplied in advance. [Cohen & al., 1989] monitor the execution of the Phoenix agents' plans and use three mechanisms for handling unexpected events: low level reflexes to stabilize the situation, error recovery and replanning implemented as high level cognitive actions, and envelopes as a general monitoring mechanism. The agent always prepares for the same fixed set of reactions, without considering the characteristics of the plan or of the situations that might be encountered during its execution. These systems have limited flexibility since the set of reactions is limited, always the same, and always available in its entirety to the execution components.

Hardware implementations of reactive plans into agents whose actions are guided by overall goals have been proposed in [Nilsson, 1988; 1992]. Continuous actions are modeled using T-R trees (teleo-reactive, i.e. both goal-directed and ever-responsive) to build a reactive program whose execution

produces circuits to control the agent's actions. Selective reactions would be very important here because of the various costs associated with hardware implementations.

The next step on the research path towards agents with better response performance was to devise techniques which shift some of the system's activities from planning to reaction, with the aim of producing increasingly reactive agents. [Mitchell, 1990] combines reactive (stimulus-response) and search-based architectures to control autonomous agents. Explanation-based learning techniques [Mitchell & al., 1986] are used to extract rules (condition-action pairs) from plans to make the Theo-Agent increasingly reactive by learning plans into reactions: the agent first tries to react, then to plan. Scaling issues for the approach are briefly mentioned, and a solution is proposed based on selective learning invocation using a utility function similar to the one suggested in [Minton, 1990]. However, as we mentioned before, there are too many characteristics of the situations and contingencies as well as of the agent (planning and execution modules) which are not taken into account by this utility function. This fact is even more important since rules are tested in sequence for reaction, which yields a high cost of reaction at execution. [Martin & Allen, 1990] propose a two-level architecture consisting of a strategic planner (generating high-level goal descriptions) which sends commands to a reactive system which must fill in the details. They use statistics to constrain the probability that the execution module can accomplish a particular task. Reactive behaviors are learned selectively, using statistical estimates on the utility of these actions versus the utility of their components. But once learned, the reactions are always available to the execution system. Soar [Laird & Rosenbloom, 1990] also provides a combination of reactive execution and planning seen as essential behaviors of an autonomous intelligent agent. Plans are learned into reactions using chunking, and afterwards all reactive plans learned are always available to the executor. The authors express their concern that after learning too many such reactions, the responsiveness of the system may be significantly reduced, but do not attempt to address this problem.

These works concentrate mainly on how to prepare reactive responses and tend to use them in such a way as to substitute regular planning. Our approach differs from these others in its recognition of the complementary

strengths and weaknesses of the two modes, and in its full integration of planning and reacting within a single agent. A recurring, unaddressed problem in these works is the value (utility) of reaction. While we believe that learning such reactions is very useful in real domains, we also believe that this utility problem should be addressed at planning time, and not (only) at learning time. The work described in this thesis is aimed precisely towards this goal. In the next chapter, we will define a framework to select only the relevant events associated with a given situation. Reactions to them are incorporated into stored reactive plans, depending on several factors such as event criticality, reaction time allowed and exhibited, load of the agent's reasoning capabilities and other resources, and reactive plan size, as well as on the desired behavior pattern for the agent. Our main problem is to decide which contingencies to prepare reactive responses for, in each situation. This is in contrast with most of the research cited above, where the authors are concerned mainly with improving their systems' performance by trying to react (and maybe cache) the most frequently used plans. Our selection criteria will necessarily be much more complex than the utility measures proposed so far.

However, the utility of reacting versus planning can also be, and has lately already been, addressed at execution time. [Horvitz, 1989] develops a decision theoretic framework to reason about the value of continuing to reflect about a problem vs. taking an action to try to solve it, at execution time, using the expected value of computation (EVC) as fundamental measure. He attempts to optimize behavior under resource constraints by integrating reaction with deliberative reasoning (replanning). However, he ignores the overhead of retrieval of a reaction and the computation time while taking into account only limited other resource constraints (e.g. memory cost) which may not be the most relevant ones for real-world agents. He also assumes all reactions are always available and only attempts to decide, at execution time, whether to react or to replan, and is not concerned with such decisions at planning time (clearly, some contingencies do not allow time for such metalevel deliberations at run-time, before taking an action to respond to them). [Yamada, 1992] uses the notion of success probability to determine the best time until which dynamic replanning may continue and when execution

of the action should actually start. Again, the computation is done at execution time.

The sixth category of techniques which we have identified at the beginning of this section involves methods to decide, at planning time, on which contingencies to select for preparation of reactive responses in the plan, and which to ignore and leave for dynamic replanning at execution time if such a contingency will arise. The problem is occasionally mentioned in the literature, but without being analyzed in detail and especially without proposing any solutions to it. While discussing the CIRCA system, [Musliner et. al., 1994] make the most comprehensive presentation of the problem that we were able to find. They recognize the limitations that exist in the agent execution resources, and attempt to divide the main plan into smaller pieces and create reactive plans that guarantee the achievement of critical goals. However, there is no analysis of how to partition the set of goals into guaranteed and unguaranteed ones (when the system cannot guarantee responses to all of them). CIRCA only tries to build guaranteed plans by taking into account only the time allowed to respond to a contingency. Other contingency characteristics relevant for the decision process (like criticality and probability) are mentioned as necessary to be considered in future works, but they are not actually used here. Control level goals are linked to the system's safety, which is not always necessary (in our work, any change in the environment that was not expected as a result of executing the main plan is considered a contingency). CIRCA also partitions the goals into just two subsets according to a system designer specified priority: critical or not.

We are unaware of any previous research towards a solution to the general problem of deciding whether to prepare a reactive response to a contingency or not; therefore, it is here where the work described in this thesis has been concentrated.

As shown before most research to date is concerned either with employing only one of the planning or reacting control modes, or simply attempts to turn a system to become increasingly reactive and rely as little as possible on planning. All the reactive responses are always available to the agent executing a plan, and they usually tend to take precedence over the (re)planning alternative. This approach can only work in either very simple

task environments, or for idealized, unlimited resource agents. In our work, we take into account the real-world constraint of limited resources for agents that have to act in stressful, resource-demanding, real-time situations, in which reaction does not come for free. Therefore, we assume that the importance of regular planning makes it irreplaceable, but the agent's performance can be significantly improved by selectively preparing reactive responses only for those contingencies that are critically enough to justify them. We work towards integrating planning with reaction, instead of just enabling the agents to shift from planning to reaction. [Hayes-Roth, 1993] proposes a paradigm for integrating planning and reaction using opportunistic control of action: run-time control conditions trigger a subset of possible actions, strategic plans constrain intended actions, and the match between possible actions and strategic plans controls action execution.

Other work, directly related to various subsections of the thesis, are briefly surveyed when relevant.

Chapter 3
The Approach

In this chapter we describe our framework for deciding, at planning time, whether to prepare a reaction for a given contingency in a certain situation. We first define a few terms which we will frequently use:

- a *plan* (*conditional plan*, or *main plan*, or *conventional plan*) is a (possibly conditional) time dependent, partially ordered set of actions and expectations (figures 2.1 and 3.1.a).

- an *action* is the application of an operator to the current state. It yields a new state, which may be identical or not to an expected state.

- a *contingency* is any state of the world entered by the executing agent while following a plan, which is not: (i) a direct consequence of executing the actions of the plan up to that point, or (ii) an exogenously generated state of the world assumed in the design of the plan. Therefore, a contingency does not necessarily affect the agent or the plan execution, and when a contingency does affect the plan, it is not necessary that it will negatively affect it. For example, a contingency may be a state which is not the current expected state according to the plan execution, but is a state which should have been reached along the way, after executing some more steps of the plan. The agent may detect it and use it to skip the unnecessary steps in the plan, for example in the same way as it was done with triangle tables in [Nilsson, 1984]. To

simplify the exposition, we also use the term *contingency* to mean any event, fact or sign that was not expected as a result of the plan execution, and which triggers an (undesired) change in the state of the world, not expected at that time in the plan, i.e. which characterizes a state as a contingency according to the previous definition.

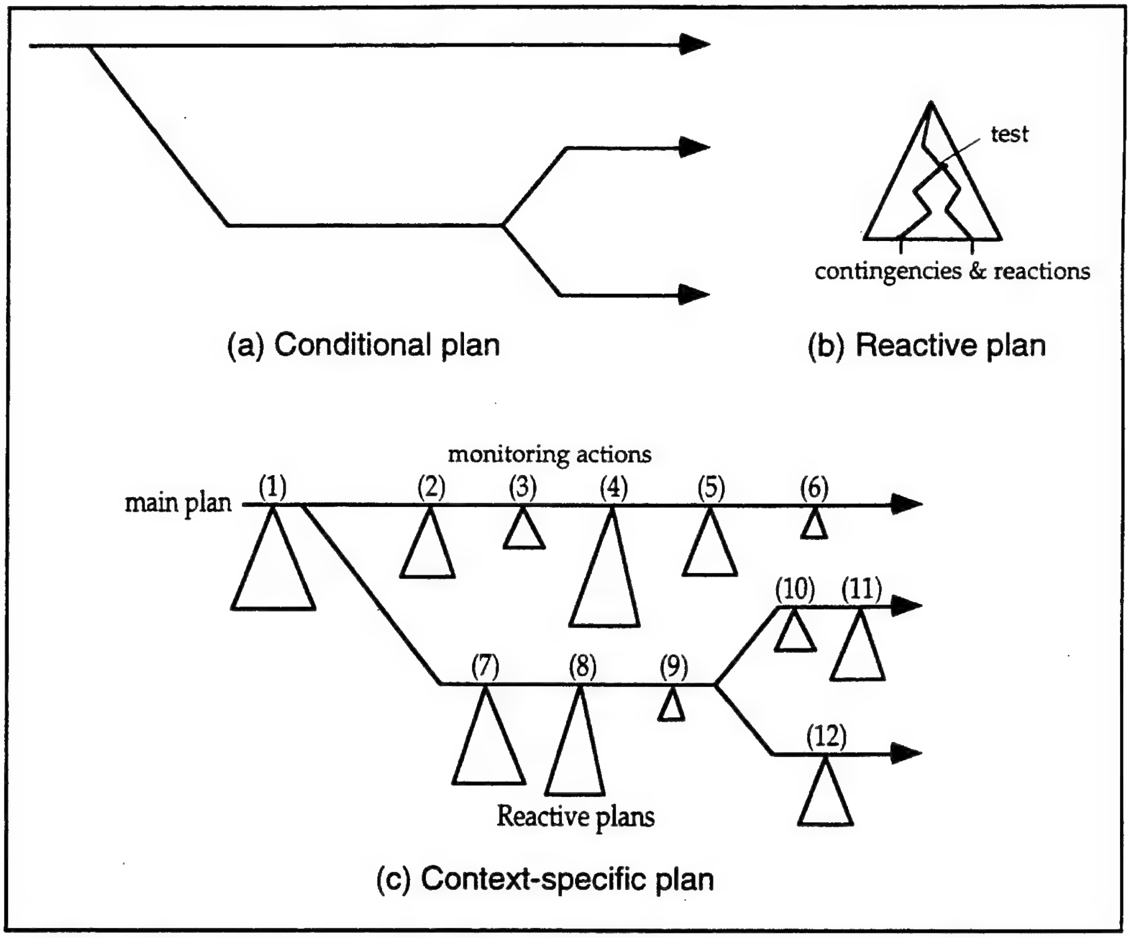


Figure 3.1. Types of plans

- a *reaction* is a perception-action rule of behavior, usually stored in a computationally efficient form. The action part may be a short sequence of actions which are enough to either solve the problem generated by a contingency, or at least to extend its deadline long enough to allow for replanning of the entire solution under the new circumstances.

- a *condition* is a pair contingency-reaction; there may be more than one reaction which can solve the same contingency, and there may be more than one contingency which can be solved by the same reaction.

- a *reactive plan* is a set of tests and reactions (possibly arranged hierarchically for efficiency reasons [Ash & Hayes-Roth, 1993] and therefore represented as triangles in figure 3.1.b) able to solve any one of a set of contingencies.

- a *context-specific plan* is obtained from a conditional plan by augmenting it with monitoring actions and reactive plans for certain contingencies (figure 3.1.c). It deals with these contingencies in a local and usually incomplete way, as opposed to the conditional plan which prepares in advance for a full treatment of the possible situations that were taken into account.

The basic approach to obtain a final context-specific plan for a given problem starts with a conditional plan (produced by a conventional planner) to achieve the main goal of the problem. The agent has a knowledge base of contingencies that may appear during the execution of plans, together with proper reactions to them. After developing a plan, this knowledge is used to analyze it and to identify situations of interest, that is, those points in the plan for which the agent knows of possible contingencies and how to respond to them.

The general agent architecture to do this is briefly discussed in appendix 1. In the rest of the thesis, we assume that the agent has already decided upon such a situation and has identified the set of contingencies which may be associated with it together with their appropriate reactive responses. Now the task of the agent is to decide for which of these contingencies to actually include responses in a reactive plan which will subsequently be attached to the main plan at the appropriate place (specified by the particular situation isolated before). The context-specific plan is thus completed by augmenting the initial main plan with monitoring actions and reactive plans for the critical contingencies (figure 3.1.c). Monitoring actions can be attached to the plan even if reactions to their contingencies are not (e.g. when the contingency is important enough to be watched for, but either its likelihood of occurrence is low enough, or the time allowed to respond to it is long enough for replanning).

In the next section, we first analyze a simple problem and try to formulate an intuitive solution. We then formalize this intuitive solution in the rest of the chapter.

3.1. Intuitive Solution

Let us revisit the driving problem presented in the previous chapters, and attempt to analyze it in more detail.

In section 2.1 we formulated the problem of an agent which commutes every morning by car from home to work, and at some point A along the way it passes in front of a school while driving straight, at 25 mph. The commute takes place at a time when children are at school, or go to school. The agent knows its route well enough to know about a few contingencies that may occur while on this portion of its route. Table 3.1 lists a partial set of such contingencies, and the best reaction for each of them known to the agent. Notice that the contingencies are dependent on the characteristics of the actual situation described. Here are some of these dependencies: the contingencies depend on the type of plan used (e.g. if the agent uses public transportation, than it need not be concerned with hitting a child, since it is not in control of the car), on the action involved (if the current action would be driving on a freeway, then the likelihood of having children running in front of the car would be much smaller), on the context of solving the problem (if the same action takes place during vacation time, when that school is closed, then again the likelihood of having a child run in front of the car decreases a lot), and so on. In the next section, we rigorously define the notion of a situation, and then precisely characterize this particular situation as an example of our definition.

In order to be useful for our purpose, the notion of a situation (and its associated characteristics) must be much more rigorously specified. Also the contingencies must be expressed in some structured language in order to allow a better representation and usage (e.g. it is important whether the car moves slowly or fast, whether the child runs from left to right or from right to left, and so on). We detail these specification requirements and present formalisms

to facilitate their expression in the next three sections of this chapter and in the next chapter.

	Contingency	Reaction
1	Child runs from right, 20 m in front of car	Brake hard and steer right
2	Car crosses w/o priority 20 m in front, from right to left	Brake and gently steer right
3	Car in front stops suddenly	Brake hard
4	Cat runs across street, 20 m in front	Brake hard and steer right gently
5	Traffic light changes red 40 m in front	Brake hard
6	Tire explosion	Brake gently and do not steer
7	A deep and medium width hole detected 30 m in front	Brake hard and steer right gently
8	Airplane lands in front of car	Brake moderately hard
9	Brake malfunction light turns on	Brake gently
10	Engine overheat light turns on	Brake gently to stop the car
11	Loud radio turns on suddenly	Adjust radio volume
12	Meteor falls on the trunk of the car	Accelerate hard
13	A ball pops in the street, from the right, at 20 m in front	Brake hard and steer right

Table 3.1. Set of contingencies for the car driving domain

Our problem is to decide which of these contingencies are critical enough to require the agent to prepare in advance reactive responses for them and which should be ignored at planning time. The solution has two phases. In the first phase, the agent must order the contingencies according to the value of reacting to them; then taking into account the characteristics of the planner and the limitations of the agent's run-time resources, it must find out how many (and actually which) of the contingencies can be taken into account for reactive treatment. In order to be able to define the value of reaction to a contingency and to be then able to order the contingencies according to this value, we have to identify the characteristics of contingencies which influence this reaction value. These characteristics are defined not for a contingency alone, but for a condition (pair contingency-response) in a given situation (as seen above, these characteristics can vary from one situation to another).

One characteristic which has been recognized by earlier research (as remarked in section 2.4) is the likelihood of appearance of the contingency in that situation. We have already discussed how the same contingency may have different likelihood in different situations. Also, different contingencies may

have different likelihood in the same situation. For example, in our case, a child running into the street is less likely than encountering a red traffic light, but more likely than having a plane land on the street in front of the car.

Since reactive response is geared especially towards satisfying real-time deadlines, of special concern is the time pressure exerted by the contingency upon the agent. This time pressure (or urgency) is inversely proportional to the actual real time allowed for the agent to act in response to the contingency. Clearly, responding to the child contingency is more urgent than taking care of the radio which has just turned on by itself. On the other hand, the child running into the street and the ball popping up in front of the car at the same distance, allow for the same time of response, i.e. exert the same time pressure onto the agent.

But the value of reacting to a contingency is also determined by the gravity of the consequences presented by the contingency if no action is taken in the allowed response time. Obviously, the consequences are much more dramatic in the case of hitting a child, than if the car hits a ball.

And finally, there is one more characteristic of the conditions that has to be taken into account. This characteristic is more closely related to the response associated to the contingency, and it takes into account the possible side-effects that may be incurred if the reaction to the contingency is taken in time. For example, the side-effects of avoiding the child by braking hard (the possibility to be hit by the car following our agent's car) and steering right (the agent's car may hit the sidewalk, or a pole on the sidewalk) are the same as for avoiding the ball through the same maneuver, and can be significantly higher than the side-effects of adjusting the radio.

We assume that the agent's knowledge base contains, along with each contingency and reaction, a set of values for these characteristics (they can be obtained from experts in the domain - as we have done it, or through automatic learning methods). These characteristics have different weights in deciding upon the value of reacting to a given contingency. As we shall see, these weights are not fixed, but they are dependent on the application domain, and also on the behavior model according to which the agent acts. We shall for

now restrict our discussion to a generally accepted (by the experts in the domain) "normal" behavior, and will briefly discuss other types of behaviors in section 6.3. Under this behavior model, the highest weight is associated to the time pressure characteristic, followed by consequences and then likelihood. However, if the side-effects are much higher than the consequences, then the agent is probably better off by ignoring the contingency at planning time.

Therefore, a driving agent will give highest priority to the child running into the street contingency (since the time pressure is very high, and the consequences are also very high), and will give a very low priority to the ball contingency, since the side-effects of doing a dangerous maneuver outweigh by far the consequences of hitting the ball. The traffic light turning red contingency will follow the child one, followed in turn by the airplane landing and the loud radio turning on (since both have low likelihood, but the airplane has much higher consequences and time pressure). The contingencies listed in table 3.1 are actually ordered according to the normal behavior model described by a panel of experts whom we have interviewed (section 6.1 presents more details about our knowledge acquisition process for this domain). At first glance it may be surprising, for example, that the ball contingency was placed after the radio contingency; remember however that we are only interested here in preparing reactions for these contingencies. Therefore, this ordering says that, if the agent has enough resources, it may try to prepare a reaction to the radio contingency (although the value of reacting to it will be pretty low), but should avoid as much as possible to prepare a reaction to the ball contingency, since the side-effects of reacting to it may be much higher than the consequences of not reacting (or equivalently, the benefits of reacting).

The second phase of our solution involves deciding which of these contingencies will actually be included in the reactive plan, by taking into account the characteristics of the reactive planner and the limitations on the agent's resources. The characteristics of the reactive planner (specified as a reactive planner model) allow the agent to estimate the complexity of isolating the contingency and its reaction from the reactive plan prepared for the entire set of selected contingencies associated with that situation. This complexity is direct proportional to the time needed by the agent from the

moment it detects the existence of a contingency and until it can start a reaction to it. However, this time is further influenced (i.e. increased) by the availability and limitations of the agent's resources, specified by an agent model (e.g. computational overhead). For each contingency included in the reactive plan, this response time has to be smaller than the time allowed by the contingency before the (re)action has to be taken (otherwise the reaction to that contingency becomes useless). Therefore, given the reactive planner model and the agent model, we have to analyze each contingency associated with the situation, in the order specified by the first phase of our analysis. In our example, we will always include in our reactive plan a response to the child contingency, since it has the top priority. We will also include in the plan a response to the car crossing contingency, if we estimate that the agent will have the resources to react to both contingencies in time, and so on. If we reach a contingency which cannot be responded to in the allowed time period while still being able to respond to all the contingencies included in the reactive plan before it, then this contingency will be left out. However, this process continues until all contingencies have been examined, since some contingency further down the list may allow a longer response time, while still allowing time to respond to all the already included contingencies. For example, assume we have time to respond to only two contingencies with very high time pressure, and to some other contingency with much lower time pressure. Then we will want to include the child and car crossing contingencies (which are the first two on our ordered list), ignore the car stopping and cat crossing contingencies for which we do not have time to respond, and include the red traffic light contingency which follows in the list, because it allows for a much longer response time. Such a policy (which is rigorously defined in section 3.4) makes optimal use of the agent's execution time resources, as justified in chapter 5).

In the following three sections we define our framework, along the lines of the intuitive analysis presented here, and in chapter 5 we make a brief analysis of some of the theoretical properties of this framework. In chapter 6 subsequently then present a few more examples of applying this framework in other domains like anesthesia and intensive care monitoring.

3.2. Framework for Reaction Decision

In the following sections we define our framework, along the lines of the intuitive analysis presented above. We specify a consistent framework to help decide whether the agent should prepare in advance to react to certain possible contingencies, or whether it can ignore them at planning time and can replan at execution time to deal with them. As seen before, the inclusion of monitoring actions and/or reactive responses for a particular contingency in a plan may depend on a large number of characteristics of the environment, the contingency and its response, and on the relations between them, as well as on the models of the different factors involved in this process: the expert, the agent and the reactive planner. They also depend on the set of other contingencies possible in the same situation (how many, how critical, and how complex their reactions are) vs. the agent's capabilities. To help visualize the heuristic rules that take these decisions, we define a few multi-dimensional spaces and the relationships among them. The position of a contingency in these spaces determines whether or not the agent reacts to the event.

3.2.1. Overview of the Framework

We begin with a general presentation of the interactions among the components of our framework, and in the subsequent sections we present in detail each of these components.

Figure 3.2 presents a schematic overview of the framework described here. The entire framework is used to decide, for a given condition (pair contingency-reaction), whether the agent should include the reaction to this contingency in the reactive plan which is prepared for the situation under consideration. Therefore, given the condition and the situation, the framework has to provide the means to associate a criticality value to the contingency. This criticality reflects the value of reacting to the contingency (using its associated reaction, if it appears in this situation), as opposed to leaving the agent unprepared to respond to this contingency and hoping that it will be able to solve it by dynamic replanning if the need will arise. If the reaction value is high enough, the agent will at least monitor for the occurrence of this contingency during execution of this phase of the plan.

However, the agent may not be able to prepare for all contingencies with criticality high enough to be monitored for.

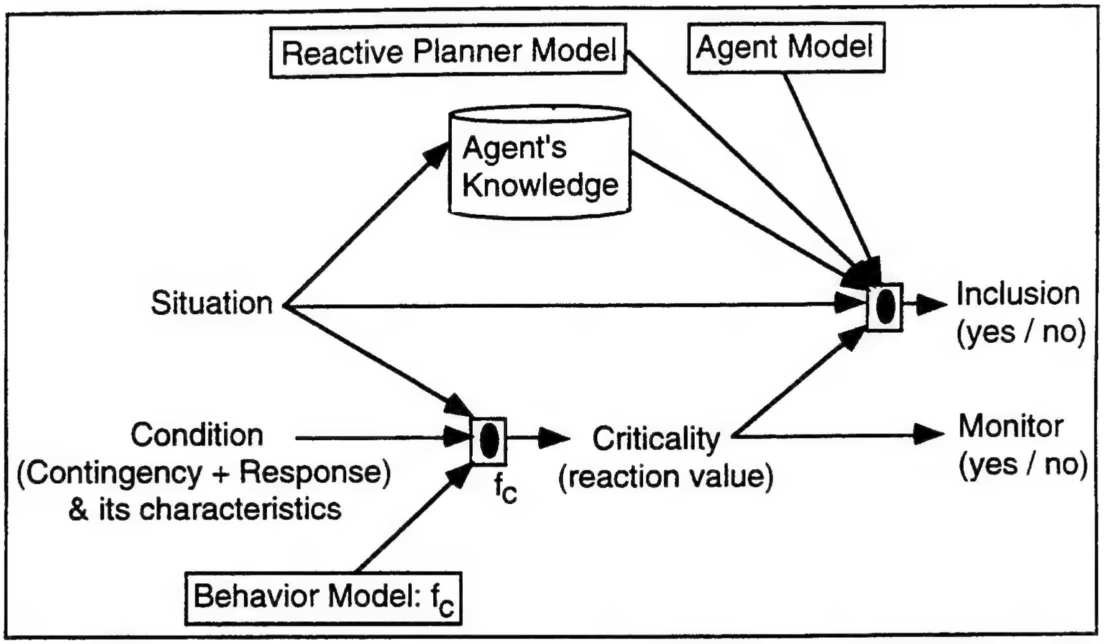


Figure 3.2. Overview of the Framework

The decision of whether to include the reaction to this contingency in the reactive plan is taken based on the characteristics of the situation, the time pressure exerted by the contingency upon the agent (or equivalently the time allowed for response by the contingency), and of course the criticality of the contingency, compared with the criticalities of the other contingencies known to the agent to possibly appear in the current situation. The criticality values induce an order relation on the set of contingencies associated with a situation, and the agent first attempts to include the most critical of these contingencies for reactive response. All the contingencies (taken from the agent's knowledge base) associated with the current situation are considered in turn for inclusion, in the order of their criticality value. When reaching the stage where the current contingency is analyzed, all the contingencies applicable in the current situation, with higher criticality, have been already analyzed, and for some of them (not necessarily all) the agent has decided to include reactive responses in the reactive plan. The current contingency will be included in the reactive plan only if the agent using this new reactive plan will be able, at execution time, to respond to this contingency in its allowed time, while still being able to respond in their allowed times to all the

contingencies already included in the reactive plan. In order to take this decision, our framework needs a model of the characteristics of the reactive plan built by the agent, as well as a model of the execution time characteristics of the agent resources and their limitations.

Figure 3.3 presents in more detail the source and flow of information through our framework. Each situation has a number of characteristics, and is therefore represented as a point in a situation space. This representation allows for flexible generalizations and for the representation of sets of related situations as regions in the situation space. Similarly, the characteristics of a contingency will define the dimensions of a criticality space, in which each point represents the value of reacting to that type of contingency. The third space used represents the reactive plan characteristics, in terms of the resources required by the execution of the reactive plan (given by the reactive planner model) and the resources available for execution by the agent. The agent model gives indications on how these resources are managed by the agent and how they are used by other modules of the agent, as well as the limitations on the agent resources, and is therefore used in the final stage of the decision process. The expert model is used by the framework to interpret the values suggested by the expert for the characteristics of the contingencies, and specifies a set of threshold values for these characteristics. Finally, the behavior model defines the function which computes the criticality value for each contingency. Different behavior models associate different values for the same reaction to the same contingency, according to the individual values of its criticality space characteristics. The two critical stages of the framework are establishing the criticality or reaction value of the contingency, and making the decision of whether to include its reaction into the reactive plan built for the current situation.

In the remaining subsections of this section we discuss in detail each of the three spaces mentioned above, and then we present a complete summary of the entire framework. The following two sections will then describe the two critical points of the framework mentioned above.

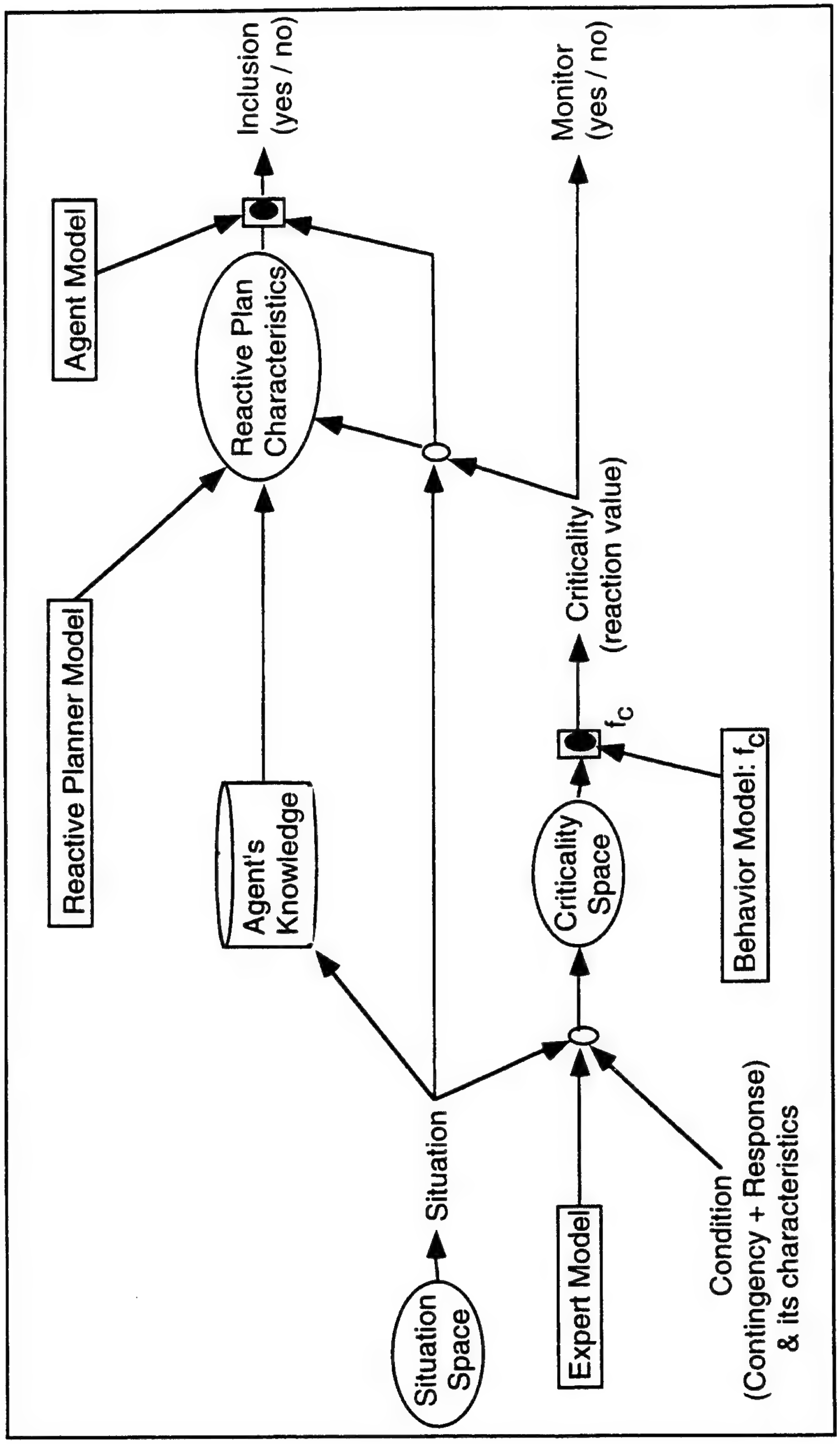


Figure 3.3. The General Framework

3.2.2. The Situation Space

The situation space is the set of all possible situations. Its dimensions are the aforementioned characteristics of a situation. A point in this space characterizes a general, contingency-independent environment situation or state. Situations will be used to index contingency-response pairs in the agent's knowledge base, according to the relevant situation characteristics in which they may apply. We will elaborate more on the same driving example used before, and will try to specify it more accurately from the perspective of our problem. The seven dimensions of this situation space are:

- *problem* - is the main problem to be solved by the agent. It is a synthesis of the problem characteristics and how they can determine the global situation. An example of problem is to carry a small package of books from home to work. We shall use this example throughout this section. A small change in the problem statement can have important influences on the set of contingencies that can be expected. For example, if the problem is instead: carry a small package of radioactive material from home to work, then an entire subset of contingencies generated by the fact that the package contains radioactive materials has to be taken into account.

- *plan* - is a synthesis of the characteristics of the type of main plan used to solve the problem. The type of plan chosen by the conventional planner is obviously dependent on the problem to be solved. For example, the plan may differ depending on the size of the package to be carried, on its weight or on its content, as well as on the distance to be traveled. However, even for the same given problem there may be a large number of solutions (plans to solve it), and each of them may create different conditions with which contingencies may be associated. For example, for our problem, one can choose to walk or to use a means of transportation, and further, to drive or to use public transportation, and further to drive a car or a bike, or any combination of these, and so on. Let us assume the planner's choice was to drive a car.

- *context* - is a synthesis of the characteristics of the environment in which the plan is to be executed to solve the problem. It covers all the general aspects of the domain which are not covered by the previous two dimensions. For the driving example, it includes the time of the day (it may make a considerable difference for the types of contingencies to be expected, whether it is day or night), the time of the year (in winter, the road is usually more slippery, but the engine is less likely to overheat), weather conditions, the abilities of the driver, and so on. Suppose in our example the context is a working day morning during the month of May. This means that children are going to school, and therefore children and balls can be very well expected into the street around the school.

- *action* - is the action to be currently executed by the agent according to the plan. Since the contingency preparation process is an off-line analysis of the main plan, "current" here means the currently analyzed time point of the plan. Non-execution of planned actions (missing actions) may also be represented on this dimension, since contingencies may occur both associated with the execution of actions in the main plan (e.g. steering to the right may cause the car to slip sideways) as well as with non-execution of an action (e.g. not steering to the right when the road turns right may have obvious consequences...). In our example the action is just to drive straight ahead on street S at a speed of 25 mph.

- *expectations* - are descriptions of situations (changes in the state of the environment) along the plan path. In order to monitor the execution of the plan, the agent looks for some important such states which are prespecified at planning time. We call these states *milestones*. The achievement (or not) of a milestone may determine the agent to change the conditional plan branch which it is following, and therefore to change the set of contingencies for which it is on the lookout. According to the way they may be generated, there are two kinds of expectations which must be taken into account when defining a situation:

- *internal expectations* - due to actions performed by our agent while executing the plan (e.g.: an attained milestone may be entering on a freeway, as expected, while to the contrary, an unattained milestone may be a situation in which the agent did not enter the freeway, although this was expected as a result of executing a set of plan steps). Such an occurring state change can be foreseen, and if the change does not occur, it becomes a contingency: it may signal that something went wrong with the plan execution, and therefore the agent should try to find out what and replan, but in the meantime it should be on the lookout for a certain set of contingencies that may also appear in this situation. For example, due to driving on street S, the agent expects (as milestone) to arrive in front of a school. If it does not, then maybe the plan was not entirely correct and the agent is somewhere else than it should be at that time. It should therefore react (attempt to stop) and replan: attempt first to find out where it is (e.g. by reading the street signs), and then replan its route from there on.

- *external expectations* - due to other independent agents which work in the same environment (e.g. changes in traffic lights). These agents may generate contingencies by themselves, since they actively change the environment; their actions may have a certain non-zero degree of correlation with the actions of our agent, or may be totally uncorrelated. For example, the traffic light is an agent whose actions may be somewhat correlated with our agent's actions if our agent approaches the traffic light from some direction where there are street sensors or other traffic lights synchronized with this one; otherwise, the traffic light's actions are totally uncorrelated with the actions of our agent. Two kinds of events may be distinguished here too: (i) something may happen (like the signal change) or (ii) something expected may not happen (e.g. a malfunctioning red signal which does not change after a long waiting time period). In the example situation we have been building in this section, a possible external expectation might be to notice children in the area (since it is a working day morning in May and we are in front of a school). However, this is not a

milestone: it is possible that the children may be in class at that time, and this fact does not alter in any way the execution of our main plan.

○ *time* - this basic characteristic of planning problems will appear in each of the abstract spaces we consider, although with different meanings (when the possibility of confusion arises, we will denote the time dimension for the situation space with $time_S$). Here it represents the amount of time elapsed since some action was taken or since a situation change was noticed, or the amount of time allowed until a situation change must appear. It is therefore strongly coupled with the expectations dimensions (expectations become more or less stronger with time passage). For example, if we allow for 3 minutes from the moment we start driving on street S until reaching the school and the expectation is not met, then something wrong may be going on (e.g. a traffic jam, or a deviation from the route) and the agent should try to replan (or maybe first to react and then to replan) for an alternate route.

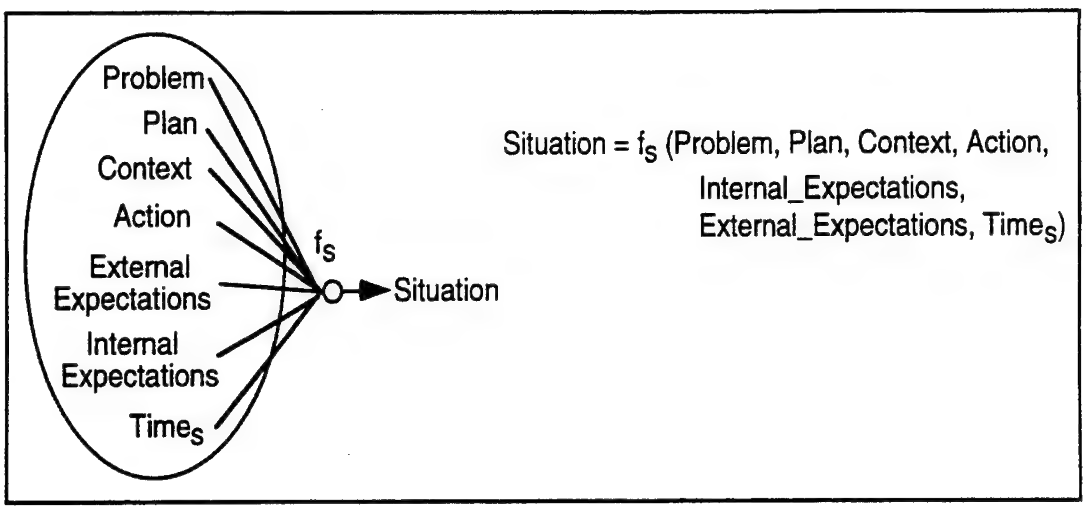


Figure 3.4. The Situation Space

The values along each dimension of the situation space are descriptions of those dimensions, as given in the example built during this section and summarized in section 3.2.5. A point (called *situation*) of this space, fully defines (for our purposes) the agent's situation, that is: the action executed and the current expectations in the course of executing a certain type of plan to

solve a given problem in a specific general context or environment. We will use it further to determine whether the agent should prepare or not a reaction for a contingency "in the current situation". In chapter 4 we present a representation formalism for the values of the situation space dimensions, which allows us to group situations into classes to facilitate the storage of knowledge and the reasoning and knowledge acquisition processes for an agent using our framework. Figure 3.4 summarizes the functional dependencies described here.

With each point in the situation space, there is a (possibly null) set of contingencies (and responses) associated (known to the agent through its knowledge base) for which the agent has to further decide whether to watch for and to prepare reactions for. Let us suppose that the contingencies known by our agent to be associated with the situation described in this section are the ones listed in table 3.1. However, we shall mainly discuss and compare the characteristics of only two of these contingencies, which have essentially the same reaction: (i) children running in the street in front of the car, and (ii) a ball appearing in front of the car. As the need will arise, we will refer to other contingencies in the set for comparisons too.

3.2.3. The Criticality Space

The criticality space describes the characteristics of a contingency and its associated reaction in a specific situation, and helps in establishing the value of performing the reaction when the contingency appears in that situation. In the previous subsection we used the situation space to evaluate a situation, independently of the contingencies that might appear in it. Here we evaluate the criticality of a contingency, dependent on the situation in which it occurs, but independent of the set of other possible contingencies for the same situation, and independent of the characteristics of the reactive planner and those of the agent. Resuming our driving example, we continue to exemplify our presentation by analyzing the two contingencies associated with the situation described during subsection 3.2.2. The four dimensions (with situation-dependent values) defining the criticality space are (figure 3.5):

- *time* - is the time deadline, or the urgency to correct the problem raised by the contingency. This is in contrast with the time dimension for the

situation space introduced in the previous subsection, which represented the time allowed to pass until a contingency is declared. We actually use two strongly correlated values here:

- *Time*$_{rc}$ - is the actual real-time interval allowed to pass (without consequences) between the time a contingency is detected and until the corrective action is taken.
- *Time*$_{p}$ - is the corresponding time pressure acting upon the agent; it is inversely proportional to the real time (the proportionality factor is a parameter of the expert model).

In our example, in both the child and the ball case, this is the dynamic planning time available before the action must be taken in order to avoid collision, from the moment the contingency is detected. This time is shorter than, for example, the time allowed to respond to the radio turning itself suddenly loud. Therefore, the time pressure is much higher in the first two cases than in the radio contingency.

○ *consequences* - is a summary of the gravity of the consequences that may appear if no action is taken (before the time deadline) in response to the contingency. This value can (but need not) be situation dependent. In our example, hitting a child can be fatal, and this value will be very high. But hitting a ball is usually no big deal, so its value will be small.

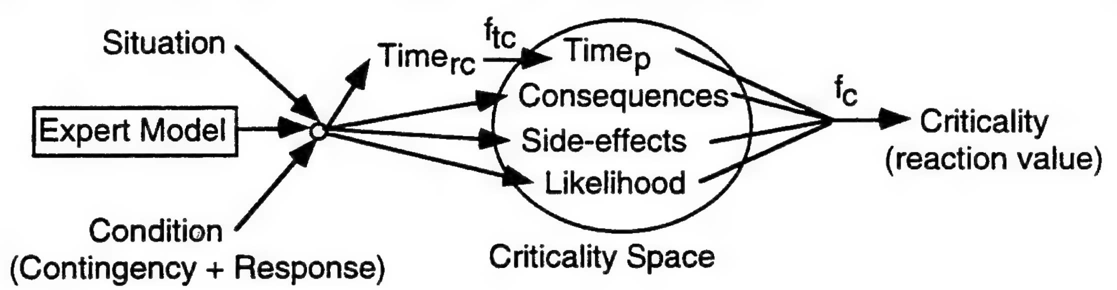
○ *side-effects* - is a summary of the gravity of the consequences that may occur as a result of reacting, and therefore this characteristic is mainly dependent on the reaction and the situation, and less dependent on the actual contingency. Alternatively, it is a measure of the risk of not being able to reach the final goal anymore, once the reaction is executed. In our case, in order to avoid hitting the child or the ball when driving a car, the same reaction is indicated. It is a dangerous maneuver (braking hard implies the possibility to be hit by the car following our agent's car, and steering right implies the possibility that the agent's car may hit the sidewalk or a pole on the sidewalk) and this yields a high value for the side-effects characteristic in this case, i.e. significantly higher than, say, the side-effects of adjusting the radio.

○ *likelihood* - this dimension summarizes the probability of occurrence of a given contingency in a given situation. However, it is important to note that it need not be the actual probability, or not even perfectly correlated to it. It can simply be a value that is approximately correlated to the actual probability, in that the relative values of the probabilities of different contingencies are reflected in their relative likelihood values. Initially, this value can be determined from previously known cases in the literature describing the domain, from the estimates of an expert, or from a theoretical analysis when a sufficiently strong domain theory exists. Later on during its lifetime, the agent may adjust it according to its own experience. Assuming the agent has no prior experience in our example, we initialize the likelihood as medium for both a child and a ball appearing in front of the car passing in front of a school, with the likelihood for the ball contingency a little higher than for the child one. They are both higher than the likelihood to have an airplane land on the street, but lower than the likelihood to encounter a red traffic sign.

The values along the consequences, side-effects and likelihood dimensions of the criticality space are reals in the interval [0,10]. The values for the time pressure dimension are real numbers greater than 0; the upper limit for the time pressure depends on the threshold values imposed by the expert model, which will be discussed in section 3.3.1. All the values for all the criticality space dimensions may be specified qualitatively (e.g. for the consequences dimension using {very small, small, medium, high, very high}) and are then translated into numeric values. These values are situation dependent; they may be different for the same contingency associated with different points in the situation space. For example, the side-effects of the proposed dangerous maneuver to avoid a collision with a child or a ball are much smaller if driving in an empty, large parking lot, than when driving on a busy street. The values for the criticality space dimensions for each condition and situation must be specified in the agent's knowledge base. It is important to note here that these values need not be very precise in absolute values. It is enough if they are in the correct order and approximately of correct relative values. This is because the method for computing the criticality value (section 3.3.2) and the way this value is used further in the

framework are robust (i.e. noise tolerant), making the entire framework robust. We shall substantiate these remarks in chapter 6, when we shall discuss the experiments we have conducted. Given these relaxed precision requirements, the experts with whom we have worked on the knowledge acquisition part of our experiments were able to specify quickly and with little effort suitable values for the characteristics of the contingencies in our experiments.

A point in the criticality space presented here defines an expected value for the reaction to a contingency, versus a dynamically replanned response, as shown in section 3.3.2. The agent attaches to the plan such a reaction only if the contingency is critical enough with respect to the other contingencies possible in this situation, and only if it will have enough resources at execution time to respond in time to this contingency as well as to all the previously accepted contingencies. That is, as we shall see in section 3.4, not all such reactions *can* be included, but monitoring actions for all contingencies found to be critical enough (according to an expert defined threshold) after this analysis will be included in the plan.



$Time_{rc} = f_1$ (Situation, Condition) Consequences $= f_2$ (Situation, Condition)
Side-effects $= f_3$ (Situation, Condition) Likelihood $= f_4$ (Situation, Condition)
$Time_p = f_{tc}(Time_{rc}) = k / Time_{rc}$ Monitor $= f_m$ (Criticality)
Criticality $= f_c$ ($Time_p$, Consequences, Side-effects, Likelihood)

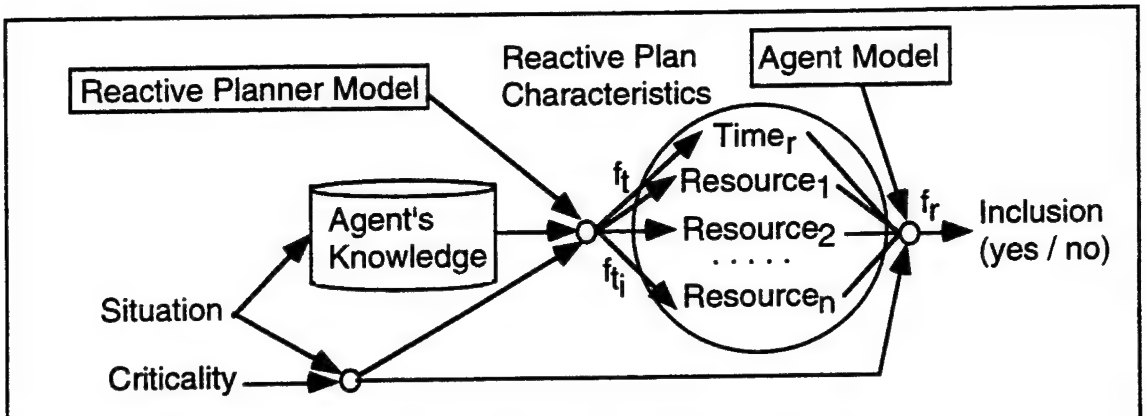
Figure 3.5 The Criticality Space

Figure 3.5 summarizes the characteristics of the criticality space defined above, and their relationships (functions) to other elements of our framework. Functions f_1 to f_4 are implicitly contained in the expert model; they are not explicitly used in the framework, since the values for the four

dimensions of the criticality space are acquired directly form the experts. However, for well-structured domains, it is possible that a strong domain theory might exist which can explicitly specify these functions.

3.2.4. Reactive Plan Space

The reactive plan characteristics represent one more set of features to consider in deciding whether to prepare a reaction to a contingency or not. We define a reactive plan characteristics space to help us study the relationships between replanning a response, versus reacting to the same contingency in the same situation. The factors to be taken into account here are the availability of computational and non-computational resources of the agent, expressed through the reactive planner model and the agent model (subsections 3.4.1 and 3.4.2). Here, the values of the dimensions in this space will be based on all the elements of our framework: situation, contingency criticality, and reactive planner and agent models. Thus, we have built our framework hierarchically, the coordinates of each space of the framework being defined in terms of the values of elements in (and the dimensions of) the previous spaces.



$Time_r = f_t$ (Situation, Criticality, Agent's_knowledge, Reactive_planner_model)

$Resource_i = f_{t_i}$ (Situation, Criticality, Agent's_knowledge, RP_model)

Inclusion = f_r ($Time_r$,$Resource_1$,...,$Resource_n$,Agent_model,Situation,Criticality)

Figure 3.6. Reactive Plan Characteristics Space

The dimensions of the reactive plan space, which also represent the characteristics of reactive plans, are (figure 3.6):

- $time_r$ - is the time needed by the agent between the moment a contingency is detected, and until the proper reaction to it can be started; it depends on both the computational and non-computational resources of the agent, their capabilities and their load in that situation. The value of this dimension grows with the number of the contingencies included in the reactive plan and with the complexity of identifying them and their reactive responses.
- $resource_i$ - is the total requirement imposed on the agent's i-th resource by the reactive plan containing the current contingency analyzed plus all the contingencies previously decided to be included for reactive response and associated with this same situation. These dimensions are of special concern for real systems. Both computational and non-computational resources (including memory) are limited, and their availability may be decisive for the successful completion of the reaction (e.g., in the limit, a universal plan for a real domain may require an infinite amount of memory, which is unacceptable in real systems).

Inclusion of a reaction to a new contingency depends on the size of the resulting reactive plan, which combines it with the set of all the reactions to contingencies already decided to be included in the reactive plan for that situation. These contingencies were obtained from the agent's knowledge base where they are indexed by their applicable situations, and have been previously analyzed by this framework (since their criticality must be higher than the criticality of the currently analyzed contingency).

The agent's knowledge base includes all the contingency-reaction pairs known to the agent, indexed by the situations in which they may appear, and with associated descriptions for the criticality space dimensions. We shall present in chapter 4 a formalism to construct languages for representing situations, contingencies and reactions in the knowledge base, designed to take advantage of the regularities of the application domain.

To continue with our example, the more contingencies (selected from the 13 contingencies given in table 3.1) are included in the reactive plan, the more likely it is to decrease the responsiveness of the agent to each of the

contingencies included. Since we have no information (yet) on the structure of the reactive plan built by the reactive planner, and also on the agent's resource limitations, we cannot actually specify how much each of the added contingencies will increase the response time (we shall see in section 3.4.1 that for some structures of reactive plans, adding some new contingency may, in some circumstances, not increase the response time at all). In any way, the agent will always try to include at least the reaction to the child-in-front-of-the-car contingency, and will continue to add to it as many as possible, in the order given in the table. However, it will not add a contingency if either (i) its estimated response time would be bigger than its allowed response time, or (ii) if adding it would determine the response time to any previously included contingency to exceed its allowed response time (given by the $Time_{rc}$ value of the criticality space associated with this contingency).

Figure 3.6 summarizes the characteristics of the reactive plan space defined above, and their relationships (functions) to other elements of our framework. Functions f_t and all f_{t_i} are explicitly contained in the reactive planner model and are then used in conjunction with the limitations on the agent resources defined by the agent model.

3.2.5. Summary of the Framework

The purpose of our entire framework (and of the thesis for that matter) is to keep the reactive response time and other resources for very critical contingencies within acceptable (i.e. useful) bounds, while ensuring reactive behavior at least for the most critical contingencies known for every situation. Given the information contained in the three spaces defined above, the agent has all the data it needs to be able, for every contingency, to take the decision of whether to include it or not in the reactive plan associated with a given situation. The result of processing the contingencies through the entire framework is a partition of the set of known contingencies possible in a given situation into two classes: to be included in and to be excluded from the reactive plan.

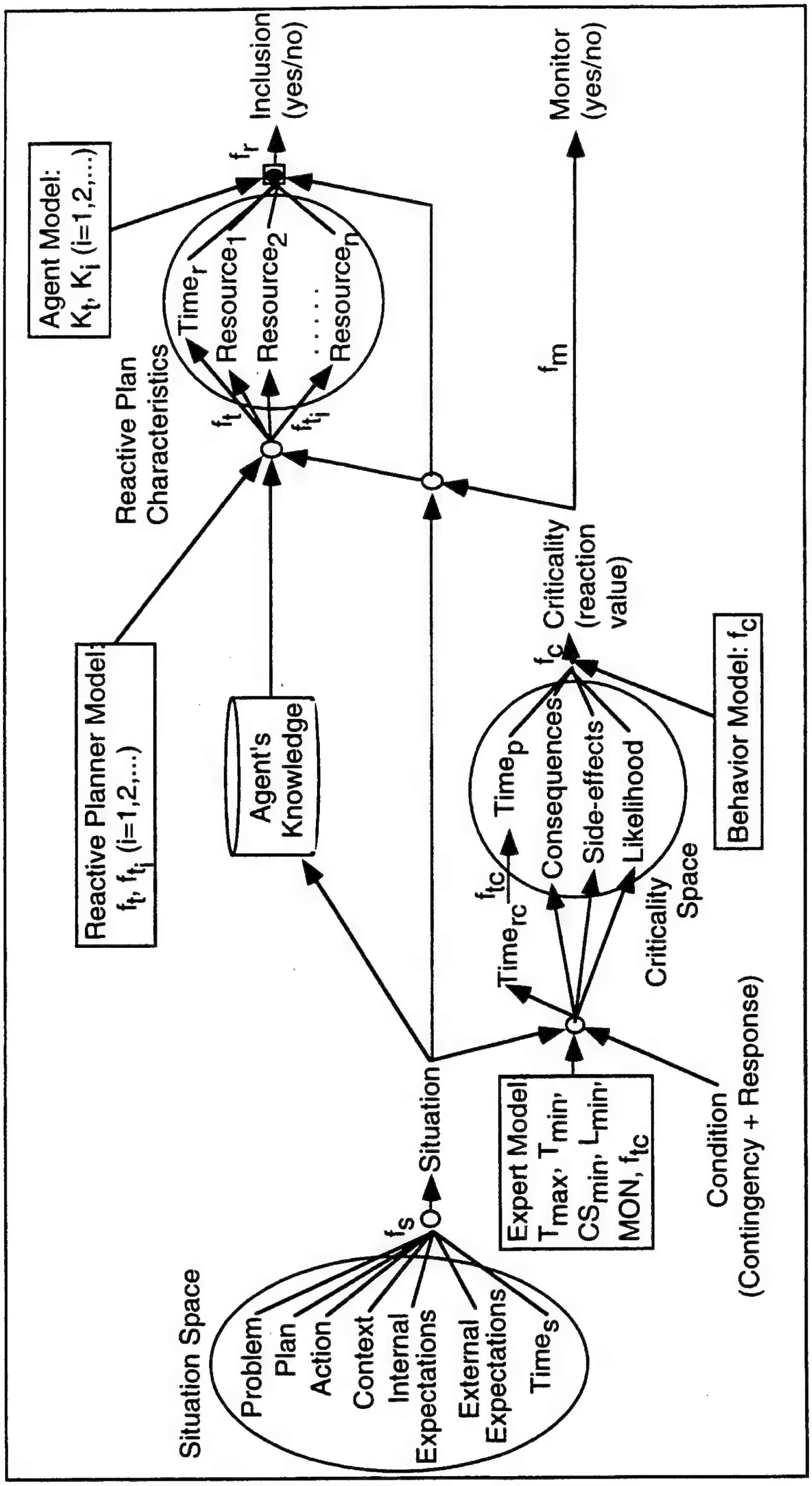


Figure 3.7. The Plan-to-React Decision Framework

Figure 3.7 shows a detailed summary of the framework for selecting the contingencies for which reactions are prepared and those for which monitoring actions are added to the plan. It details the diagram presented in figure 3.3, and essentially combines figures 3.4, 3.5 and 3.6. At any time, the agent knows of a set of contingencies and reactions to them. Each contingency may be associated with several regions in the situation space, and each point in the situation space may have several contingencies associated (many-many relationship). Each contingency is characterized in a situation by a criticality point. While the criticality value alone decides which contingencies will be monitored in which situations, the decision for including the treatment of the contingency in the reactive plan associated with that situation is made based on both the criticality value, and the reaction value of the entire reactive plan for that situation, in relationship with the reactive planner model and the agent model.

$$\text{Situation} = f_S\,(\text{Problem, Plan, Context, Action, Internal\_expectations, External\_expectations, Time}_S)$$

$$\text{Time}_{rc} = f_1\,(\text{Situation, Condition})$$
$$\text{Consequences} = f_2\,(\text{Situation, Condition})$$
$$\text{Side-effects} = f_3\,(\text{Situation, Condition})$$
$$\text{Likelihood} = f_4\,(\text{Situation, Condition})$$

$$\text{Time}_p = f_{tc}\,(\text{Time}_{rc}) = k\,/\,\text{Time}_{rc}$$

$$\text{Criticality} = f_c\,(\text{Time}_p, \text{Consequences, Side-effects, Likelihood})$$

$$\text{Monitor} = f_m\,(\text{Criticality}) \text{ - Expert Model}$$

$$\text{Time}_r = f_t\,(\text{Situation, Criticality, Agent's\_knowledge, Reactive\_planner\_model})$$
$$\text{Resource}_i = f_{t_i}\,(\text{Situation, Criticality, Agent's\_knowledge, Reactive\_planner\_model}) \qquad (i = 1,2,\ldots)$$

$$\text{Inclusion} = f_r\,(\text{Time}_r, \text{Resource}_1, \ldots, \text{Resource}_n, \text{Agent\_model, Situation, Criticality})$$

Figure 3.8. Functional Relationships for the Plan-to-React Decision Framework

The set of functional relationships among the elements of the framework is summarized in figure 3.8.

Appendix 1 presents the general agent architecture and the basic data flow during the plan modification process.

Our agent integrates reactive responses with the plan to compensate for the unfeasibility of universal plans. It does not only try to prepare for the most frequent or likely contingencies, but also for some very infrequent ones which are very critical. Due to real-world resource limitations, some of the frequent but not very critical contingencies may be excluded from reaction in favor of less frequent but very critical ones.

Space	Dimensions	
Situation	Problem	Deliver package to work
	Plan	drive car
	Context	school time (May, week)
	Action	drive straight, 25 mph
	Intern. Expectations	reaching school
	External Expectations	children in sight
	Time	max. 3 mins.
Contingency		Child / Ball in front of car
Criticality	Time	to avoid collision (short)
	Consequence	fatal (very high) / small
	Side_effects	high
	Likelihood	medium
React. Plan Characts.	Time	N.A. / to be considered
	Memory	N.A. / to be considered

Figure 3.9. Example for the driving domain

Two advantages of the framework introduced here are: (i) its specification is general, domain and agent-independent, so we expect it to be applicable to a wide variety of agents working in a variety of environments, and (ii) it is highly parameterized, which ensures a proper adjustment of the framework to a specific agent and to domain-dependent requirements

(domain, expert, reactive planner, and agent characteristics and capabilities)[1] as well as to the desired type of behavior. In chapter 5 we claim and justify that the framework, as presented here, is free of redundancies; that is, each of the elements included in our framework are necessary to completely describe the characteristics of a contingency and its reaction in order to allow the agent to decide at planning time whether to prepare for the reaction to that contingency in that situation. While we cannot prove that the framework is also sufficient (i.e. that there are no other elements needed for this decision besides the ones described here), the experiments described in chapter 6 were successfully conducted using this framework. Should the need to extend the framework arise, we believe that it can be easily done, while preserving the elements and their structure discussed here.

Space	Dimensions	
Situation	Problem	inguinal hernia
	Plan	surgery procedure H
	Context	heart disorder history
	Action	apply anesthetic
	Internal Expectations	get patient asleep
	External Expectations	surgeon perf. incision
	Time	from action to sleep
Contingency		heart failure
Criticality	Time	to restore heart (short)
	Consequence	fatal (very high)
	Side_effects	very low
	Likelihood	high
React. Plan Characts.	Time	N.A. (irrelevant)
	Memory	N.A. (irrelevant)

Figure 3.10. Example for the anesthesia domain

[1] In a specific setting (domain, expert, reactive planner and executing agent), these parameters can be automatically or interactively learned using paradigms like the ones proposed in [Dabija, 1990; Dabija & al., 1992a,b].

Figure 3.9 presents a summary of the car driving example used throughout this section to illustrate our framework. Figure 3.10 presents an example from a different domain - anesthesiology, to show the generality of our theoretical framework.

The agent is an anesthesiologist preparing for an operation during which contingencies that endanger a patient's life may appear. The situation space is defined by the general characteristics of the operation (inguinal hernia to be treated through a specific surgery procedure performed on a patient with heart disorder history). The plan analysis is at the point where anesthetic is applied. This action will give rise to two kinds of expectations (milestones) to be watched for: as a result of the action, the patient should get asleep after a certain amount of time, and from the external environment the expectation of an incision being performed by a surgeon. At this point, the anesthesiologist agent analyzes as a possible contingency a heart failure. It has a short deadline (the time to restore the patient's heart without causing brain damage) and the consequences of not reacting in time are fatal (very high). It also has a high likelihood of occurrence, given the patient's medical history. As we shall see in the following sections, since these characteristics yield a very high criticality value for this contingency, the agent will probably decide to add monitoring actions to the plan, and will probably include its reaction in the reactive plan for this situation, almost regardless of the rest of the contingencies relevant to the same situation (analogous to the child contingency in the driving example). In chapter 6 we present a larger set of results which we have obtained from our experiments in this medical domain.

3.3. Establishing the Value of Reaction

As mentioned in the overview of the framework which we made in section 3.2.1, our framework has two critical phases: establishing the criticality (or reaction value) of the contingency, and making the decision of whether to include its associated reaction into the reaction plan built for the current situation. In this section we will concentrate on the first of these phases, and will leave the second one for the next section. But before we can present our method for establishing the reaction value of a contingency, we

have to talk briefly about the expert model, since it is according to such a model that the values for the criticality space dimensions are specified.

3.3.1. The Expert Model

The situation-dependent criticality space values for a contingency-reaction pair are supplied by an expert, and are thus subject to the personal interpretation of the expert, according to his own *expert model*. As our experiments have shown (chapter 6), the experts need not be very precise in the absolute values they provide. It is enough if they are in the correct order and approximately of correct relative values. This is because the method for computing the criticality value (section 3.3.2) and the way this value is used further in the framework are robust (i.e. noise tolerant), making the entire framework very robust. We shall substantiate these remarks in chapter 6, when we shall discuss the experiments we have conducted. Given these relaxed precision requirements, the experts with whom we have worked on the knowledge acquisition part of our experiments were able to specify quickly and with very little effort suitable values for the characteristics of the contingencies in these experiments.

The values specified by the expert for each contingency are the real time interval allowed between the moment a contingency is detected and until its reaction is started, the consequences of not reacting to the contingency, the side-effects of executing the reaction associated with the contingency, and the likelihood of occurrence of the contingency in that situation. The last three values are real numbers in the interval [0,10]. The values for the time pressure dimension are positive reals; the upper limit for the time pressure depends on the threshold values imposed by the expert model, which are presented below. All these values may be specified qualitatively (e.g. for the consequences dimension using {very small, small, medium, high, very high}) and are then translated into numeric values (e.g., corresponding to the previous set of qualitative values, these numeric values will be in the intervals: {(0.2], (2,4], (4,6], (6,8], (8,10]}. As seen in previous chapters, these values are situation dependent; they may be different for the same contingency associated with different points in the situation space.

The expert model reflects the expert's interpretation of the domain (and the way he or she estimates the values of the contingency characteristics). This model must include the following threshold values, which will be used in the next section in our analysis:

- T_{max} - is an upper limit on the reasonable values for the time pressure exerted by contingencies on the agent. A time pressure higher than this value makes the reaction useless since it can only be taken too late (the agent has no way to react before the deadline). In our driving example, the meteor contingency has a too short deadline to be responded to realistically, so the agent is better off by not including such a reaction in the reactive plan (and leaving the reactive plan only for contingencies that can be responded to in reasonable time).

- T_{min} - is a lower limit on the time pressure values for which the agent should try to respond reactively. If the agent has more time than this threshold, then it can probably dynamically replan its response, thus leaving room in the reactive plan for other, more time pressuring contingencies. Therefore, the value of reacting here is significantly lower, although not zero - if the agent has left enough execution resources, then maybe it is still a good idea to prepare a reactive response for such a contingency. For example, if the agent driving a car detects a traffic jam, it does not have to react (well, usually...) but can take its time to replan an alternate route. However, we can easily imagine traffic jam situations in which the agent is much better off by first reacting (and, say, leave the freeway) and then replanning, than just by taking its time to dynamically replan (and, say, pass the freeway exit).

- L_{min} - is a lower limit on the likelihood of occurrence of contingencies for which the agent should prepare reactions. A likelihood value lower than this threshold indicates that the contingency is so unlikely to appear in this situation that the overhead of preparing and managing a reactive response is probably unjustified, so the value of reacting here is significantly lower. An example here can again be the meteor contingency, and maybe the airplane landing contingency too. This treatment can be dangerous in certain domains where the

consequences may still be fatal, but in such cases this threshold can be lowered to zero. Also, the value of reacting if the likelihood drops below the threshold is again still positive (though much smaller), so if the agent has left enough execution resources, then it may again be a good idea to prepare a reactive response for such a contingency.

- CS_{min} - if the side-effects of a reaction to a contingency outweigh the consequences of not reacting by more than this value, then it is probably wiser not to take any action. In this case, like in the upper time pressure threshold, T_{max}, the value of reacting to the contingency is considered zero. An example is the contingency of a ball popping up in front of the agent's car: the side-effects of taking the recommended dangerous maneuver outweigh by far the consequences of hitting a ball at 25 mph, so the agent is better off by ignoring this contingency from the reactive plan preparations.

- *MON* - is a criticality threshold beyond which monitoring actions for the contingency should be included in the main plan (even if reactions to it cannot be included); the reason is that the decision to include a reaction for a contingency is taken dependent on the agent's run-time resources and performance, which may change over time, but are not taken into account at this stage of the decision process. Also, these monitoring actions may detect a contingency for which no reactive response was prepared, but for which the agent has the resources to dynamically replan its response.

The agent model must also specify the function f_{tc} which transforms real-time values into time-pressure values. These pairs of values are inversely proportional, so this function has the form:

$$Time_p = f_{tc}\,(Time_{rc}) = k \,/\, Time_{rc}$$

where only the constant k has to actually be specified by the expert model, and has to be in some (weak) correlation with the two time pressure thresholds presented above.

Also implicitly contained in the expert's model are the functions f_1, to f_4 which associate the values for the criticality space dimensions with each pair condition-situation, as discussed in section 3.2.2.

3.3.2. Value of Reaction

The criticality value for a contingency-reaction pair is a measure of the merit of the reaction to the contingency as opposed to dynamically replanning a response to that contingency, in a particular situation in which the contingency is known to possibly appear. This value induces an order relation on the set of contingencies that can appear in that situation. This order is used to allow the selection of those contingencies that should be reacted to given the limited resources of the agent. Function f_C, which computes the criticality value for a contingency given the values of the characteristics of the criticality space for the contingency, implements the evaluation function of the behavioral model to be exhibited by the agent.

The *behavior model* represents the type of behavior which the agent attempts to simulate. By imposing an order (i.e. a preference of treatment) on the set of contingencies associated with a situation, the agent commits itself to a pattern of reactive behavior. It involves both which contingencies are preferred over which, and which contingencies are ruled out altogether from the reaction process. Each behavior model is characterized by an *evaluation function* which, given a set of conditions (pairs contingency-reaction) and a situation in which they apply, computes a score with the following property: the higher this score is, the better (more appropriate) that set of contingencies is (according to the particular reaction philosophy of that behavior model). The evaluation function orders the set of contingencies associated with a situation according to their priority for a reactive response.

The behavior model is implemented in our framework through the relative values of the parameters in the function computing the value of reaction (which is presented here), and through the values of the thresholds on the criticality space dimensions (presented in the expert model) relative to the values of the parameters of the criticality function. In chapter 5 we prove a few properties of the relationship between the evaluation function of a behavior model and the criticality function defined below. The most important

property is that both functions define the same order relation on a set of contingencies associated with a same situation, which implies that the criticality function can be consistently used to implement behavior models.

The criticality function we have used in our experiments has the following general form:

Criticality = $f_C(t, c, s, l)$ =

$$
\begin{array}{|lll}
\textit{if} & (t > T_{max}) & \\
 & \textit{then} & f_C = 0 \\
\textit{elseif} & (c + CS_{min} - s < 0) & \\
 & \textit{then} & f_C = 0 \\
\textit{elseif} & (t < T_{min}) & \\
 & \textit{then} & f_C = \sqrt{t^{P_1} * c^{P_2} * s^{P_3} * (c+s)^{P_4} * (c+CS_{min}-s)^{P_5} * l^{P_6}} \\
\textit{elseif} & (l < L_{min}) & \\
 & \textit{then} & f_C = \sqrt{t^{P_1} * c^{P_2} * s^{P_3} * (c+s)^{P_4} * (c+CS_{min}-s)^{P_5} * l^{P_6}} \\
\textit{else} & & f_C = t^{P_1} * c^{P_2} * s^{P_3} * (c+s)^{P_4} * (c+CS_{min}-s)^{P_5} * l^{P_6}
\end{array}
$$

where, for the purpose of stating the criticality function in a more succinct form, we made the following notations for the (situation dependent) criticality space dimensions:

t = $Time_p$	(is the time pressure)
c = Consequences	(of not reacting)
s = Side-effects	(of the reaction)
l = Likelihood	(of encountering the contingency)

Parameters T_{max}, T_{min}, CS_{min}, L_{min} are dependent on the domain and are defined by the expert specifying the domain knowledge. Their meaning has already been defined in the previous subsection. They are important in implementing a specific behavior model. For example, if the upper threshold on the time pressure T_{max} is made lower, than more contingencies will be left out of the reactive plan since the agent estimates that there is not enough time at execution time to give a timely response to these contingencies. This behavior simulates the resignation behavior model [FAA, 1991] (the agent leaves responses to contingencies to others, since it believes there is no use to

try to react to them, i.e. it believes that there is no time to take care of them anyway). On the other hand, taking $T_{max} = \infty$ emulates a behavior intended to avoid legal liabilities by always doing something.

Parameters p_1 to p_6 are also used to model different (human) behaviors: their relative values place the agent in different behavioral models and can be viewed as labels for human reactive behavior. For example, $p_1 > p_5 > p_6 > p_2$ (with p_3 and p_4 very low) represents what is usually accepted as normal behavior in the car driving domain: most importance is given to the time pressure and then to the difference between consequences and likelihood, with more emphasis on consequences; lastly, it also considers the likelihood of occurrence. Another behavior model in which consequences and especially side-effects are almost disregarded with respect to time pressure implements an attitude of invulnerability - the agent is prone to risk taking and does not believe that anything wrong can happen to him. Again, it is important to notice the robustness of our model: the only important thing about these parameters are their relative values, and these can themselves vary widely while still obtaining consistent results. This property makes the life of the domain experts participating in the knowledge acquisition and behavior model specification process much easier. In chapter 6 we shall discuss a number of experiments we have made and how they justify our claims for the framework robustness.

As stated before, the value of reaction associated with a contingency induces a total order relation on the set of contingencies associated with a certain situation. This is only a partial order on the set of all contingencies known to the agent, since contingencies in different situations may not (although sometimes can) be comparable according to their criticality values. This order relation is defined as:

"*A* is more_critical_than *B* " if and only if:

> *A* and *B* are contingencies applicable in the same situation S, and
> *A* has higher criticality value than *B*, or
>> *A* and *B* have same criticality, but *A* has higher consequences, or
>> *A* and *B* have same criticality and same consequences, but *A* has higher likelihood.

This ordering characterizes the behavior model of the agent. It will subsequently be used to choose the contingencies for which reactions are prepared (section 3.4.3).

Different combinations of these parameters defining the criticality function are used in both the theoretical and experimental evaluations to prove certain conjectures. In chapter 5 we claim that the parameterized function defined here can implement the human reactive behavior models described in the literature, and while we cannot formally prove this claim, we justify it through the experiments discussed in chapter 6. Therefore, our framework can also be used in psychological studies of "hazardous" attitudes in certain high-risk domains like nuclear power plant operation and airplane flying. In section 6.3 we present and briefly evaluate a series of experiments we have conducted with our framework to simulate a number of reactive behavior models described in the literature.

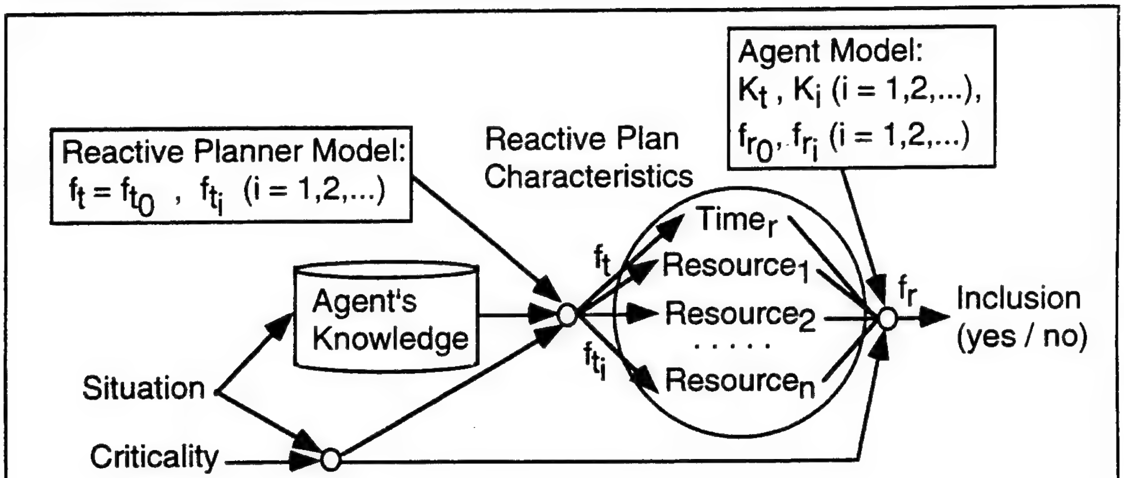
3.4. The Reaction Decision Making

Making the actual decision of whether to include the contingency and its associated reaction into the reaction plan built for the current situation is the second and last critical phase of our framework. This phase is based on all the elements and the information previously acquired and computed by the framework. As shown in figure 3.7, there are two agent dependent models that participate in this phase: the reactive planner model and the agent model. They synthesize the agent's properties and the limitations on its resources at planning time and execution time respectively. We first make a brief presentation of these models and the information they are expected to contain, and then we give the actual algorithm for deciding whether to plan to react.

3.4.1. The Reactive Planner Model

The reactive planner model describes the planning time properties of the agent, and the characteristics of the reactive plans built by the agent and their relationships to the agent's execution time resources (computational time as well as other non-computational resources). This model must allow the agent, at planning time, to estimate the variations in execution time resource

requirements with respect to the growth of the reactive plan, namely with the number of contingencies and reactions included in the reactive plan. This is accomplished by the functions f_t and f_{t_i} in figure 3.11 which depicts the entire decision making process presented in this section.



Time$_r$ = f_t (Situation, Criticality, Agent's_knowledge, Reactive_planner_model)

Resource$_i$ = f_{t_i} (Situation, Criticality, Agent's_knowledge, Reactive_planner_model)

Inclusion = f_r (Time$_r$, Resource$_1$,..., Resource$_n$, Agent_model, Situation, Criticality)

Figure 3.11. The Reaction Decision Making Phase

Function f_t estimates the time needed by the agent from the moment it detects the existence of a contingency and until it can react to this particular contingency, when the reactive plan known to the agent in this situation contains the response to this contingency as well as responses to all the contingencies with higher criticality which apply in the current situation. The reactive planner model assumes that the agent can devote all its computational resources to this task (this assumption is then taken care of by the agent model, described in the next section, which takes into account any overhead that the agent may experience in that situation). Function f_t estimates how much does the reactive response time increase, on average, by adding this contingency to the reactive plan.

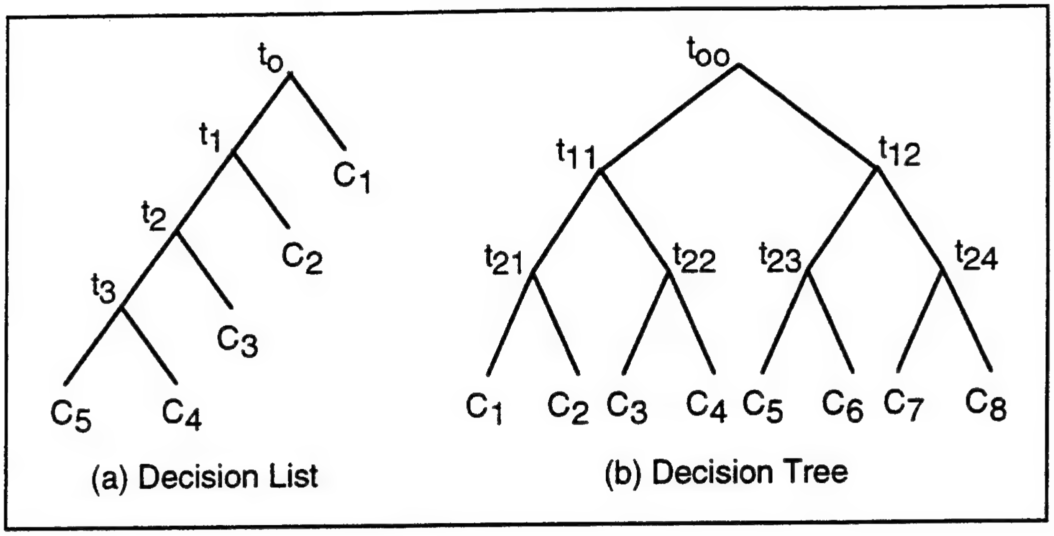


Figure 3.12. Two reactive plan models

Two commonly encountered examples of reactive planner models are decision lists and decision trees. For a reactive planner based on decision lists (figure 3.12.a), the time to react increases approximately linear with the number of contingencies to be considered, since for each new contingency added to the reactive plan, a new test must be added to discriminate it. Therefore, the time needed to react to a contingency according to this model will be the sum of the times required for each test that has to be done before deciding on the contingency. If we assume the testing time to be roughly constant, then the estimated time to react becomes:

$$\text{Time}_r = \text{test\_time} * \text{rank\_in\_reactive\_plan}$$

i.e. is directly proportional to the number of tests to be performed which is equal to the number of levels in the decision list before the contingency in question. In figure 3.12, t_i $(i = 0,\ldots,3)$ and t_{ij} $(i = 0,1,2;\ j = 0,\ldots,4)$ are tests to be performed in order to determine the proper reaction to the contingency, and C_i $(i = 1,\ldots,8)$ are the contingencies (and their associated reactions) for which the reactive plan contains responses.

If the reactive planner uses decision trees to index the reactions in the final reactive plan, then the time to reach a response is closer to the logarithm of the number of contingencies (the base of the logarithm is equal to the branching factor (assumed constant) of the decision tree), assuming again an approximately constant testing time. Figure 3.12.b presents such a complete

binary tree, for which the reaction time for each of the contingencies is roughly:

$$Time_r = test\_time * \log_2 (number\_of\_contingencies\_in\_reactive\_plan)$$

i.e. is directly proportional to the logarithm of the number of contingencies treated by that reactive plan (we assume complete decision trees, in which the k leaves (contingency-reaction pairs) are all situated at level m if $k = 2^m$, or 2p of the leaves are at level 2^m and the other k-2p leaves are placed at level 2^{m-1} when $k = 2^{m-1} + p$, $(1 < p < 2^{m-1})$.

Similar reactive planner models can be built for other methods of organizing the reactions in reactive plans.

Functions f_{t_i} have the same mission for each of the other critical resources of the agent (e.g. the amount of memory needed by the reactive plan, as well as any other non-computational limited resources that the agent might need in order to start its reactive response), as f_t has for computational time.

The two formalisms for structuring reactive plans mentioned above (complete binary decision trees and decision lists) deserve here a brief comparison. At the first glance, a qualitative reasoning seems to imply that decision trees are better (or at least never worse) than decision lists. After running the experiments described in chapter 6, we have found out that this is not necessarily the case. We shall show here when this is not necessarily true, and analyze and justify it. (A formalism is considered better if it can include more reactions to more critical contingencies in the reactive plan to be executed by the same agent with the same resource characteristics and limitations, in identical situations). During this discussion we will assume that all the tests require the same amount of time (T), and that there are enough tests available such that any arrangement of reactions in the respective reactive models is possible. In this case, responding to the n-th contingency in the reactive plan will take time $T * n$ in the decision lists case, and $T * \log_2 (n)$ in the case of complete binary decision trees.

We must note two things here: (i) different contingencies may have significantly different time pressures (i.e. significantly different allowed

response times), and (ii) a structural difference between decision lists and decision trees is that the complete decision tree takes the same amount of time to respond to all the contingencies, while decision lists respond faster to contingencies placed towards the root of the list, and this response time increases with the distance of the condition from the root.

Therefore, once the decision tree reactive planner has decided to include a given contingency (say C) in the reactive plan, it can only add so many contingencies to the plan until the estimated response time to contingency C becomes larger than its allowed response time. This means that the decision tree formalism is actually limited by the contingency with the highest time pressure which the agent decided to include in the reactive plan. This is not the case however for reactive planners based on decision lists. Here, the contingencies with the highest time pressure can be placed towards the root of the tree, and the response time to them will not be affected by the number of contingencies covered by that reactive plan. Therefore, contingencies with lower time pressure can still be added towards the end of the decision list, since they allow for a longer time of response, and will not affect the response time for contingencies placed higher on the list. A number of experimental results which support this analysis (actually, as we stated earlier, they have prompted this analysis) are presented and discussed in section 6.2.

In summary, when the response times allowed by the contingencies under consideration vary within a small relative range, the decision tree based reactive planner will be able to include more such contingencies (since all its leaves are reached in roughly the same amount of time). On the other hand, when the time pressures of the contingencies vary widely (which tends to be the case in real-world domains), decision lists are better suited for including responses to a larger number of contingencies, since testing first for contingencies with shorter time of response allows timely reactions to more contingencies with lower time pressure. Naturally, the best solution would be an incomplete decision tree which combines the advantages of both formalisms.

In this thesis, we assume that the agent has enough planning resources and time to build the most comprehensive reactive plans which do not exceed

its execution time resource limitations. However, this framework may also be applied when dynamically replanning courses of actions, and when the limitations on the agent's planning resources needed to build such reactive plans may become a factor to be considered. In such cases, the reactive planner model may also be required to estimate the complexity of the reactive plan structuring algorithm. This estimate can then be taken into account by our framework, and may lead to the decision of reducing the set of conditions to be included into the reactive plan, in order to ensure that the time required to construct the reactive plan will not exceed the time allowed for this task.

3.4.2. The Agent Model

The second agent dependent model involved in this later stage of the framework in which the agent makes the actual decision of whether to include the contingency and its associated reaction into the reaction plan built for the current situation is the agent model. It synthesizes the agent's properties and the limitations on its resources at execution time.

The agent model describes the (situation dependent) response capabilities of the agent (figure 3.11). The functions (fr_i) describe the variation of the availability of resource i (i=0 for computational time) due to the fact that the agent cannot devote its entire $resource_i$ exclusively to responding to that contingency. For example, the computational load on the agent slows its responsiveness by a factor K_t greater than 1, and can be expressed by:

$$fr_0\,(time_r) = time_r * K_t\,;$$

or if the agent can devote itself to solving this contingency only after some constant time K_a, then

$$fr_0\,(time_r) = time_r + K_a,$$

and so on.

The agent model also supplies the amount of each resource (K_1, K_2, ...) that may be allocated to reacting in the given situation, for the non-computational resources. Example of non-computational resources are, in the anesthesiology domain, oxygen masks and ventilators. Such resources are

available in limited quantity, and also may only become available after a certain waiting period. The agent model does not have to specify such an upper limit on the availability of resources for computational time, since this is already specified separately for each contingency through the reaction time allowed to respond to it (the time pressure dimension of the criticality space values associated with the condition in the agent's knowledge base).

The agent model is very important in domains where non-computational resources may not be available all the time, but may be obtained after some waiting period (as in medical domains like anesthesia or intensive care monitoring, or in nuclear power plant operation).

By comparing the requirements of each of the agent's run time resources, for the set of the previously included contingencies plus the current contingency under consideration, with the limitations on the availability of that respective resource (given by the agent model for non-computational resources and the agent's knowledge base for time), the agent can decide whether this contingency can be included in the reactive plan for the current situation or not. We shall analyze this decision process in detail in the next subsection.

3.4.3. Deciding Whether to Prepare to React

The final purpose of this entire framework is to decide, for each contingency-response pair associated with a given situation, whether to preplan the reaction to it or not. As shown in figure 3.11, this decision is taken by comparing the estimated execution time resource requirements for the agent to respond to all the contingencies already decided to be included in the reactive plan plus the contingency currently under consideration, with the allowed response times for each of these contingencies in that situation.

Given the criticality of the current contingency and the set of the other contingencies known possible in the current situation, this decision process proceeds as follows: the framework computes the agent's execution time resource requirements to respond to any of the contingencies as:

$$\text{Resource}_i = f_{t_i}\ (\text{Situation, Criticality, Agent's\_knowledge, RP\_model})$$

for each resource_i (i = 0,1,...) of the agent (for a unitary exposition we shall sometimes call the agent's computation time as resource_0; all other resources of the agent (possibly including the amount of memory needed by the reactive plan, as well as other domain dependent critical and limited resources like ventilators in an intensive care unit, etc.) are numbered starting with 1. The functions f_{t_i} are given by the reactive planner model, and estimate the increase in resource_i requirements by adding this new contingency-reaction pair to the reactive plan. For i = 0, $f_{t_0} = f_t$ estimates how much does the reactive response time (considered from the time a contingency is detected, and until a reaction to resolve it can be taken) increase, on average, by adding this condition to the reactive plan. As discussed in subsection 3.4.1, f_t is approximately linear for decision lists and roughly logarithmic for decision trees. Obviously, the better the reactive planner model is (i.e. the better these estimates are), the better use of the execution time resources of the agent will be ensured by the selected set of contingencies.

As we have mentioned in section 3.3.1, the decision to monitor for a contingency is taken based only on the criticality value of the contingency, and independent of the reactive plan characteristics. The reason is that the decision to include a reaction for a contingency is taken dependent on the agent's run-time resources and performance, which may change over time, but are not taken into account for monitoring purposes. Also, these monitoring actions may detect a contingency for which no reactive response was prepared, but for which the agent has the resources to dynamically replan its response. The decision to monitor is taken as a threshold function on the criticality of the contingency:

$$\text{Monitor} = f_m\,(\text{Criticality}) = (\text{criticality} \geq \ \text{MON}) =$$

$$\left|\begin{array}{llll} \textit{if} & (\text{criticality} \geq \ \text{MON}) & \textit{then} & f_m = \text{yes} \\ \textit{else} & & & f_m = \text{no}\,. \end{array}\right.$$

where MON is the monitoring threshold defined by the expert in the expert model (section 3.3.1).

The final decision of preparing a reaction for the currently analyzed contingency is taken by the function f_r:

React = f_r (Time$_r$, Resource$_1$,...,Resource$_n$, Agent_model, Situation, Criticality)

=

$$
\left|\begin{array}{llll}
\textit{if} & \text{criticality} < \text{MON} & \textit{then} & f_r = \text{no} \\
\textit{elseif} & f_{r_0}(\text{Time}_r) > \text{Time}_{rc} & \textit{then} & f_r = \text{no} \\
\textit{elseif} & f_{r_1}(\text{Resource}_1) > K_1 & \textit{then} & f_r = \text{no} \\
\textit{elseif} & f_{r_2}(\text{Resource}_2) > K_2 & \textit{then} & f_r = \text{no} \\
\;\;\ldots\ldots & & & \\
\textit{elseif} & f_{r_n}(\text{Resource}_n) > K_n & \textit{then} & f_r = \text{no} \\
\textit{else} & & & f_r = \text{yes}\,.
\end{array}\right.
$$

$$= (\text{monitor} \wedge \prod_{i=0}^{n} (f_{r_i}(\text{Resource}_i) \leq \ K_i)) \ ,$$

where resource$_0$ is the real computational time, and K_0 = Time$_{rc}$ is the real response time allowed by the contingency for the response to be started without consequences (the time pressure dimension of the criticality space values for this contingency).

The functions f_{r_i} are given by the agent model, and describe the execution time overhead imposed by other processes which the agent has to attend to in the same time in which it must respond to the contingency. Equivalently, they describe the availability of resource$_i$ for this reactive plan. They may be therefore situation dependent, and can be described as such in the agent model. A common expression for these functions is of the form:

$$f_{r_i}\ (\text{resource}_i) \ = \ \text{resource}_i * k_t + k_a \ ,$$

where k_t is the overhead due to the agent's load (or the portion of it which can be expressed as a delaying factor), and k_a is an initial delay or cost associated with the use of that resource (for example, a process which cannot start before a certain lead time, or a resource which cannot be delivered to the agent before a waiting period has elapsed). All these parameters must be specified by the agent model.

```
// input: a situation
// output: a list of reactions (symptoms-actions pairs) for that situation

cr-list <- extract from the agent's KB all contingency-reaction pairs matching situation;
    // cr-list is the set of all the contingencies known to the agent to be possible in situation

for each contingency in cr-list do
    time-pressure <- f_tc (time_rc);                                          // expert model
    criticality <- f_c (time-pressure, consequences, side-effects, likelihood); // behavior model
    if criticality > MON
        then monitor <- true
        else monitor <- false
    if not monitor
        then eliminate this contingency from cr-list
enddo

cr-list <- order cr-list by criticality value, then by consequences, then by likelihood

include <- ()
    // include is the set of all the contingencies to be included in the reactive plan
    //        associated with situation

for each contingency in cr-list do
    time_r <- f_t (include + contingency, situation) ;
    resource_i <- f_t_i (include + contingency, situation);
    inclusion <- f_r (time_r, resource_1, ..., resource_k, time_rc, k_1, ..., k_k)
        // f_r returns true iff there are enough resources to respond reactively to all
        // contingencies previously added to the list include and to the currently
        // considered contingency.
if inclusion
    then add contingency to include
enddo

return the list include.
```

Figure 3.13. Reaction decision making algorithm

```
function fr (timer, resource1, ..., resourcek, timerc, k1, ..., kk)
    // returns true iff there are enough resources to respond reactively to all contingencies
    // previously added to the list include, and to the currently considered one.

if fr0 (timer ) > timerc
    then return NO;

for i = 1 to number_of_agent_resources do
    if fri ( resourcei ) > ki
        then return NO;
enddo

for each contingency in include do
    if fr0 (contingency.timer ) > contingency.timerc
        then return NO;

    for i = 1 to number_of_agent_resources do
        if fri (contingency. resourcei ) > ki
            then return NO;
    enddo
enddo

return YES.
```

Figure 3.13. Reaction decision making algorithm (continued)

One final set of parameters specified by the agent model are the execution time resource limitations of the agent (K_i , i = 1,2,... , in the formula for f_r above). They do not include $Time_{rc}$ which is a characteristic of the contingency and therefore is specified in the agent's knowledge base. Thus, what the decision function does is simply to check that:

(i) the contingency is critical enough to be at least monitored for,

(ii) the agent will have enough time at execution to respond to this contingency in the context of the larger set of contingencies considered for reactive response in the same situation,

(iii) none of the execution time limitations of the agent resources (besides computational time) may be exceeded when attempting to respond to this contingency, considering the entire reactive plan containing it (i.e. all the contingencies with higher criticality, already decided to be included in this reactive plan), and

(iv) the agent's run time resources are still enough to respond properly to all the contingencies previously included in the reactive plan, when this new contingency is added to the reactive plan.

This decision process ensures that no reaction is included for contingencies which are not monitored for, and that there is enough available of each resource in order to attempt a reaction for all the contingencies included in a reactive plan. For the computational time resource, this means that the time needed to start a reaction to the contingency is less than the real time allowed before the action must be taken (otherwise the reaction becomes useless).

Figure 3.13 makes a brief summary of the algorithm for deciding, given a plan execution situation, on the set of contingencies to be included in a reactive plan which will be associated with the conditional plan before the actual execution starts. The actual decision function f_r is presented separately in the second part of the figure.

The fourth test mentioned above essentially repeats the second and third tests (carried out by the functions f_{r_i} , $i = 0,1,...$) for each of the contingencies already decided to be included in the reactive plan. It must be done each time a new contingency is considered for addition to the reactive plan, because the addition of the contingency, while possible from the point of view of the restrictions imposed by its characteristics, may increase the resource requirements to respond to previously included contingencies and may therefore exceed the restrictions imposed by their criticality characteristics. For example, in the case of a reactive planner based on decision trees, adding a new contingency may force the reactive planner to add one more level of tests in the decision tree, and thus increase the response time to all the contingencies included in this reactive plan. This way, some of them may now exceed the real time allowed for reaction to be taken, and their

reactions may become useless in that situation. (Conform to the analysis in section 3.4.1, the time to react to all the contingencies contained in a reactive plan with a complete decision tree structure is approximately constant and proportional to the depth of the decision tree).

The decision function f_r is applied in turn to each contingency considered for the current situation, in the order given by their criticality values, as defined in section 3.3.2 (each time, it applies each of the functions f_{r_i} , $i = 0,1,...$, to each of the contingencies already included in the reactive plan and to the current contingency, considering the reactive plan to include this contingency plus all the contingencies previously decided to be included in the reactive plan for this situation). This iterative process is continued until either all the agent's execution time resources are estimated to be exhausted, or no more contingencies are known to the agent to be possible in the current situation.

This concludes the presentation of our framework for deciding whether to plan to react. Given a plan situation and a set of contingencies known to the agent to possibly appear in this situation, it decides for which of these contingencies the agent may prepare reactive responses, considering the execution time limitations on the agent's resources. In the next two chapters we present a knowledge representation formalism to help the agent to cope with the considerable amount of knowledge related to this decision process, and theoretical justifications for some properties of our decision framework. Then, in chapter 6, we present the results of our experiments using this framework. But before doing all this, let us see how the ideas presented so far can be applied to a related problem: given a plan situation and a set of contingencies known to the agent to possibly appear in this situation, decide for which of these contingencies the agent should prepare complete branches in the main conditional plan.

3.5. Conditional Planning

We briefly discuss here how the framework presented so far for deciding whether to prepare to react to a contingency can be modified to answer the question of whether the agent should prepare in its plan a full

conditional branch for a contingency. We first resume our discussion of section 2.1 regarding possible classifications of contingencies, and then we adapt the previous framework to this new task.

3.5.1. Contingencies Revisited

In section 2.1 we have identified there types of contingencies that may appear during the execution of a plan. They are classified according to the action taken by the agent at planning time to prepare for their occurrence at execution time. These types of contingencies are:

(i) contingencies for which the planner builds complete *conditional branches*, from the contingency state to the goal state, in the main plan. As an example, suppose that the agent has two alternative routes for driving to work in the morning, depending on the color of a particular traffic light when the agent reaches it: the regular plan assumes the color is green, and the alternate branch is conditioned on the color being red. For a non-driving commuter, the plan may involve walking or taking a bus, depending on the weather, and so on.

(ii) contingencies for which the agent prepares *reactive* responses, combined into reactive plans by a reactive planner, and attached to appropriate segments of the complete plan provided by the conditional planner. An obvious example is the one we used before, involving a child running in front of the car.

(iii) contingencies ignored by the agent at planning time; their treatment at execution time can fall under two subclasses:

(a) dynamic *replanning*, if the agent has enough resources at execution time to perform it. As example, suppose that the agent encounters a traffic jam on a seldomly traveled route, for which it did not bother to prepare a conditional plan branch before execution.

(b) *noop*, that is take no action, either because the consequences of the contingencies are not high enough to warrant an action, or because the agent simply does not have the resources to take an action to solve them (e.g. they have a too short response time allowed). An

extreme example may be the contingency involving the meteor falling on the car, which we have encountered in table 3.1.

The justification for this classification is mainly related to the limited resources that a real agent can use. For a few contingencies, the agent can generate complete plans and combine them in a conditional plan. However, the agent's limited planning and execution resources do not allow for too many contingencies to be treated this way. Still, the agent can prepare at planning time reactive responses for a larger set of contingencies; these responses will not ensure full solutions to the goal state, but they will give the agent the possibility to dynamically replan its actions at execution time. But in no case can a real agent with limited resources prepare for all possible contingencies in a real world application domain. Many of these contingencies must be ignored at planning time.

Let us intuitively analyze now the characteristics of the examples given, and try to feel the qualitative differences among these classes of contingencies.

In the previous conditional planning example, the contingencies occur often, i.e. with a high likelihood (the occurrence probability may approach 50%, but should not exceed it, since if it does, then the contingency should rather be considered the normal case and the main plan should be build accordingly). Also, a solution to the contingency requires the preparation of an entire plan branch all the way to the initial goal (since the execution time may be critical and thus replanning cannot be used at any stage before reaching the goal, i.e. a local situation stabilizing response to the contingency is not sufficient), as well as certain resources whose availability must be planned in advance (e.g. an umbrella, or the correct set of maps for the alternate route to be traveled).

For the reacting case we have already devised a comprehensive framework stating the main necessary characteristics for a contingency to be considered appropriate for a reactive response. For the previous example, they include critical response time and high consequences of not responding. An important characteristic is also that a short response (already available) is

sufficient to stabilize the situation and allow for replanning of the agent's actions all the way to the initial goal.

The rest of the contingencies will be ignored at planning time, but we have been able to further subclassify them. The ones for which the agent will try to replan at execution time should not occur too often (otherwise a conditional branch may be appropriate), and should also allow for enough time for the agent to be able to build the new course of action. Finally, the contingencies for which the agent will take no action anyway (e.g., the falling meteor case) do not allow for enough time to respond to them, in any circumstances, given the agent's limited resources and execution capabilities.

In section 3.2.3 we introduced a criticality space, which is one possible representation of the space of contingencies, whose dimensions are appropriate for reaction decision purposes. To facilitate the understanding of the relationships among the classes of contingencies, we shall attempt here a simpler and more general graphical representation of the space of contingencies, which can depict all the classes mentioned above. This representation can conceptually be obtained from any more complex representation (like the criticality space mentioned before, or the importance space to be introduced later on in this section), by projecting the points in the space onto points in the simpler spaces defined here.

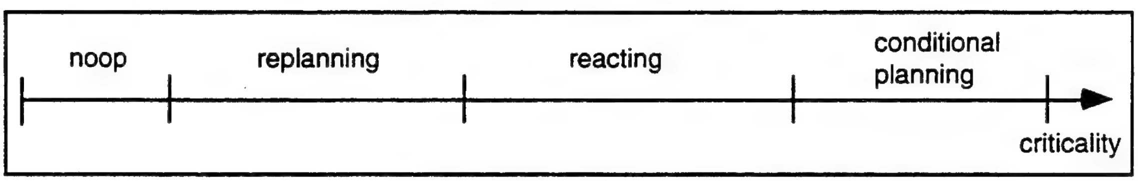


Figure 3.14. Contingency space - linear representation

The simplest representation for the space of contingencies is a linear space in which contingencies are ordered by either criticality (as defined before) or importance (as defined further in this section). Figure 3.14 shows that such a representation can outline the most frequent transitions between bordering classes, but cannot represent other still possible borderings like between reacting and noop (e.g. determined by allowed response time), or conditional planning and replanning (determined, for example, by the planning time needed). Therefore, a planar representation (figure 3.15) is

more appropriate. The dimensions here are the reaction response value and the planning response value for the contingency. While much better, this representation still does not represent the direct relation between conditional planning and noop (which, to be fair, is the least frequent one, so this representation can be used for most purposes). We have therefore devised a third, 3-D surface representation using a spherical surface (figure 3.16). The orthogonal dimensions (akin to latitude and longitude) are the same as for the second representation, and it can represent all the borders between pairs of classes.

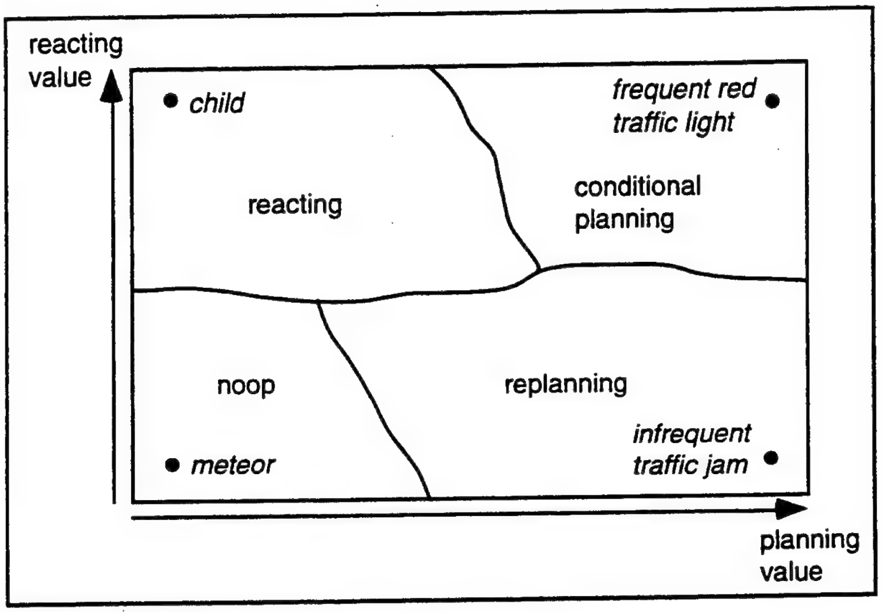


Figure 3.15. Contingency space - planar representation

The examples given with the informal description of these classes at the beginning of this section constitute extreme cases in each class (figure 3.15). In between these extreme cases there is an entire space of contingencies for which more than one (in some cases even all) of the response alternatives may be justified. The borders among these classes in the space of contingencies associated with a particular agent are determined by the agent's resource capabilities and limitations. For example, conditional planning and replanning are separated mainly by the agent's planning resources, replanning is circumscribed both by the agent's planning and execution capabilities, while reacting is mainly characterized by the agent's execution capabilities.

Due to the way the different classes of contingencies have been defined, in order to be able to best classify a given contingency, we only need class membership decision frameworks for two of them, namely conditional planning and reaction. We have already defined a framework for deciding whether the agent should prepare a reaction to a contingency in a given situation. In the rest of this section we will give a description of a framework to decide whether to prepare a conditional plan branch for a contingency in a given situation.

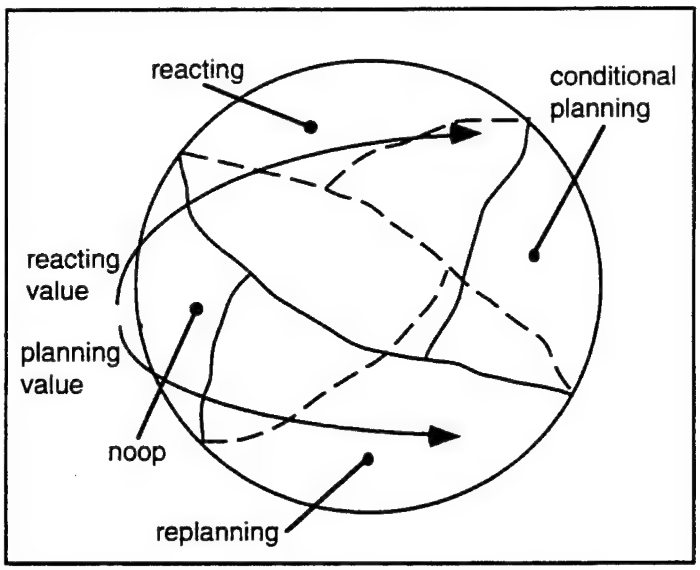


Figure 3.16. Contingency space - 3-D surface representation

There are two qualitative differences between conditional plan branches and reactions. The first is that conditional plan branches represent global solutions to the initial problem, that is, they are sequences of actions which ensure that the agent reaches the goal (in the absence of other contingencies). Reactions on the other hand are only single (or short sequences of) actions, intended only to stabilize the situation so that the agent can then take its time to replan a solution from the state reached after reacting to the initial goal. Therefore, on one hand reactions can be seen as the first steps of incomplete conditional branches, but in the same time they are more generally applicable than specific plan branches. There is also no assurance that after executing a reaction, the agent may find a plan to get it to the initial goal, i.e. it is possible that the planner may subsequently find no solution from the state in which the agent finds itself after completing the reaction to the goal; this is not the case for conditional plan branches,

assuming no other contingencies are encountered. Therefore, we always assume that a conditional planned branch is a better solution than a reaction to the same contingency, and as a consequence, given a set of contingencies for a situation, the conditional planning decision framework should be applied before the reaction one.

The second difference involves the planning process itself. In conditional planning, the planner has to work out a solution (sequence of actions) from a given state (the contingency) to the goal. In reaction planning, as assumed throughout this thesis, the agent already knows (in its knowledge base) the best reactions associated with contingencies for applicable classes of situations, so the only task of the reaction planner is to combine the reactions associated with the set of contingencies to be prepared for, into a structure which will be conveniently searched at execution time to determine the actual contingency encountered and its associated reaction (e.g. decision trees, decision lists, etc.). Therefore, planning time is definitely of importance in conditional planning, but may not be an issue when structuring a reactive plan from a set of known reactions (if it cannot be ignored, then, as mentioned in section 3.4.1, the complexity of the reactive plan structuring algorithm can be taken into account in the Reactive Planner Model, to further prune the set of contingencies for which reactions should be prepared).

Having noted these differences, we must now acknowledge that the particular decision frameworks associated with the two classes of contingencies have very similar underlying structures, so their presentations may obey the same general lines. There are significant analogies between the two problems and their solutions. They would suggest taking a unitary approach and combine the two frameworks into a single one, with aesthetical benefits of uniformity and elegance in presentation. However, we believe that this would yield an unnecessarily complex framework, intuitively difficult to present and understand. Therefore, as well as for easier understanding and to keep each framework manageable, we decided to present them separately. This is also in agreement with the way in which an agent should apply them, although in different order. Indeed, the frameworks may indicate that certain contingencies are suitable for both conditional branch and reaction

preparation. In these cases a conditional branch should be prepared, since it is assumed to be a more accurate solution, as argued before.

We first presented in sections 3.1 to 3.4 the reaction decision framework (which is the main topic of this thesis). In the remainder of this chapter we use analogies with the previous presentation to describe the conditional planning decision framework, by pointing out their similarities and differences. We transform one framework into the other by removing, adding and replacing some of its elements. Since the two frameworks are very close in form (although with underlying differences in content), an aesthetically interested reader can easily merge them together if he or she so desires.

3.5.2. Framework for Conditional Planning Decision

Let us first state the conditional planning decision problem, in a form similar to the one used in section 2.2 for reaction. We assume the agent has built a linear main plan to go from an initial situation to a given goal. The issue then is to enable the agent, for each phase of the already built main plan, to select the right set of contingencies for which to prepare conditional branches all the way to the goal. That is, the *problem* is to specify a decision framework which:

- ❍ *given*:
  - ● an intelligent agent with:
    - ✧ capabilities:
      - ✦ planning and dynamically replanning
      - ✦ monitoring
    - ✧ constraints:
      - ✦ limited resources
      - ✦ real-time performance
  - ● a linear plan by which the agent can achieve its goal
  - ● a set of contingencies known to possibly appear at certain times during the plan execution, and for which the agent may plan a conditional branch, each with:
    - ✧ known characteristics, associated with it (e.g. gravity of consequences, time deadlines) and with preplanning a conditional branch for it (e.g. resource requirements)

✧ characteristics of their replanning alternatives (replanning time and other resource requirements)

○ *enable* the agent to decide for which contingencies to prepare conditional branches in the plan (according to a desired behavior pattern) while not exceeding the agent's planning capabilities and preserving the real-time responsiveness of the agent to all these contingencies, given its limited resources.

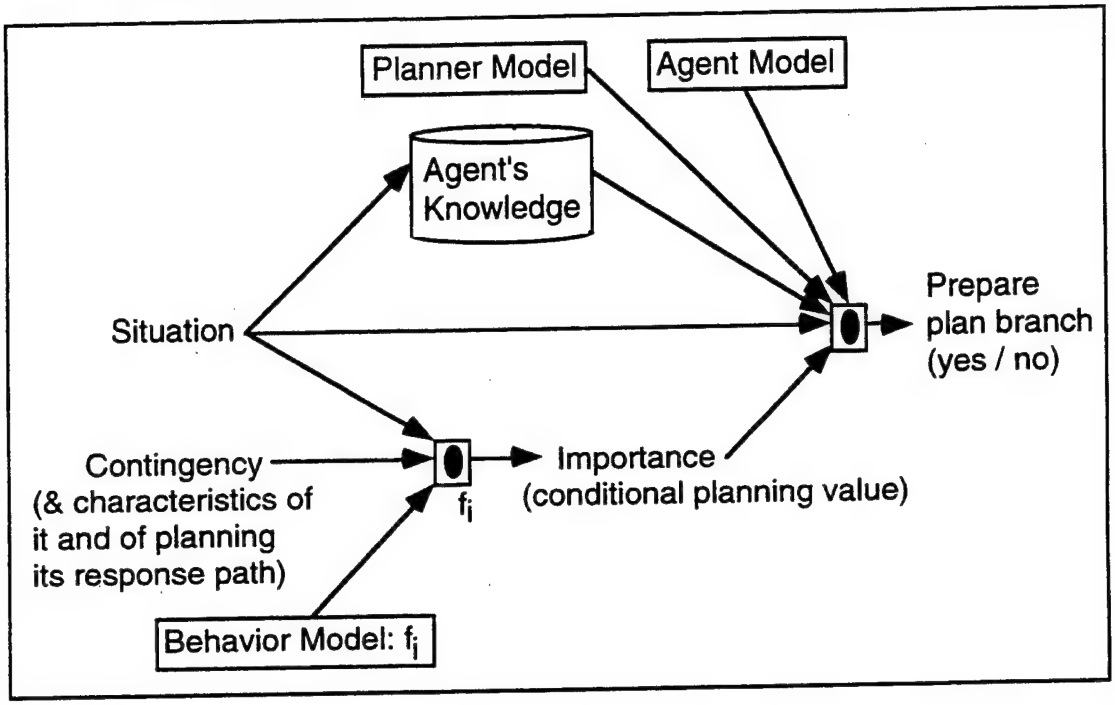


Figure 3.17. Overview of the Conditional Planning Decision Framework

As can easily be seen by comparing the two problems, they are similar enough such that a solution to the second problem can be obtained by relatively small modifications to the framework solving the first one. In fact, the high level overview of the conditional planning framework shown in figure 3.17 is very similar in form to the one for the reaction framework depicted in figure 3.2. There are however a few underlying differences to be pointed out:

○ the knowledge available to the agent and associated with the contingency does not include the response to it, but only some general

characteristics (outlined in section 3.5.3) of the planning process to be done for that contingency;

- the criticality (reaction value) computed by the reaction decision framework is replaced by an importance value (conditional planning value) which synthesizes how important it is for the agent to prepare a conditional branch for that contingency, i.e. what is the value of preparing a conditional branch for it in the plan vs. leaving it for other possible treatments;

- the reactive planner model is replaced by a model of the conventional planner used to build the initial plan and the conditional branches;

- the final decision of the framework is now whether to prepare a branch in the plan, instead of whether to include a reaction to the contingency in the reactive plan associated with it.

Also, the agent model and the behavior model will reflect slightly different characteristics in the two cases, and the functions used to calculate the conditional planning values and the final decision are based on somewhat different variables, as will become evident soon.

Figure 3.18 presents in more detail the flow and source of information through the new framework. Again, the comparison with the general framework for the reaction case (figure 3.3) shows obvious similarities between the two frameworks. The differences between the two frameworks at this level of detail and functionality are basically the same as the ones mentioned above for the higher level of abstraction used in the overview presentation.

Let us now briefly discuss each element of our new framework, and compare it where appropriate to the equivalent element of the reaction decision framework. First, the situation spaces are identical in the two frameworks, since a situation has the same definition and characteristics related to contingencies, regardless of the kind of response we prepare for them.

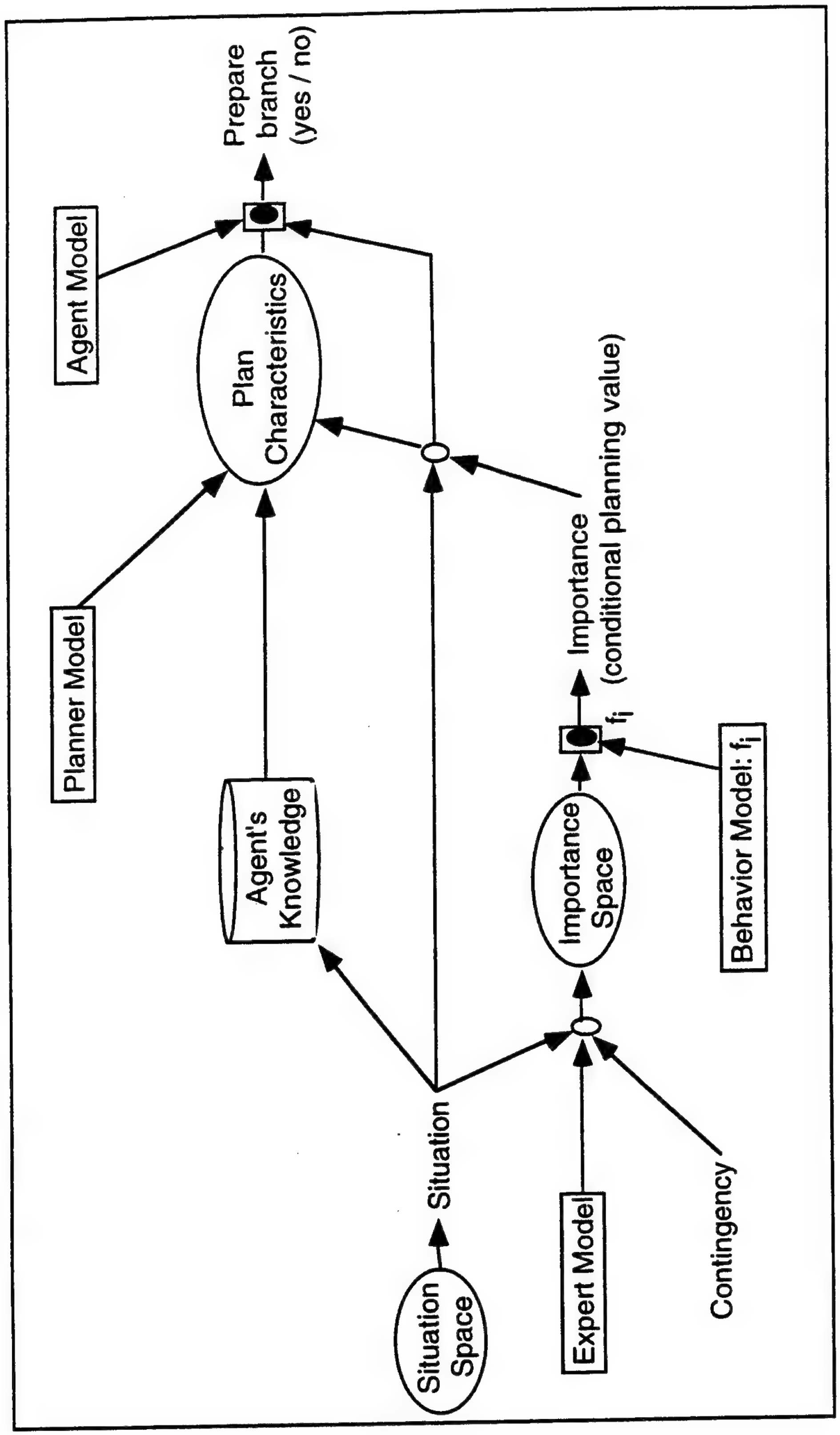


Figure 3.18. General Framework for Conditional Planning Decision

Two parts of the framework require special attention here. The first establishes the conditional planning value of the contingency, and the second takes the actual decision of whether to prepare a conditional branch for the contingency. They are briefly discussed in the following two subsections, and then we conclude this presentation with a summary of the entire framework put together.

3.5.3. Establishing the Conditional Planning Value

Figure 3.19 presents the part of the framework concerned directly with calculating a conditional planning value for the contingency in the given situation. It is similar to figure 3.5 which shows the criticality space and the process of calculating the reaction value for a contingency. We shall concentrate here on the differences between the two frameworks at this stage:

- the criticality space is replaced by an *Importance Space* which uses 5 dimensions to characterize a contingency from the conditional planning point of view. These dimensions are:
  - $Time_p$ - represents the same time pressure as in the reactive case; it is obtained from $Time_{rc}$ - the time allowed to respond to the contingency, once an unexpected state is detected (same as in the reactive case).
  - *PTime* - is the estimated planning time needed to build a branch for this contingency at planning time (e.g., the time needed to plan the alternative route, starting with a right turn at traffic light B, all the way to the office); the simplest estimate may be, for example, the planning time used to build the original plan from that point up to the goal.
  - *Consequences* - summarizes the consequences of not responding to the contingency in the time allowed (same as in the reactive case).
  - *PResources* - is a measure of how hard (time consuming, agent resource consuming and any other costs involved) it is to obtain, at replanning time (during execution) the resources needed to replan and carry out this plan branch (if not preplanned in advance).

Besides actual planning and replanning times, this also involves resources not needed in carrying out the initial plan, but which may be needed for replanning purposes (like maps which may be hard to obtain along the way) or for carrying out the alternate plan branch (like an umbrella if it rains, or in medical domains a ventilator or certain test results).

- *Likelihood* - represents the likelihood of occurrence of the contingency in the given situation (same as for reaction).

○ the *Importance* value which orders contingencies by their conditional planning value (in the same way as criticality does for reaction).

○ the function (f_i) calculating the importance value for a contingency has the form:

Importance = f_i (t, pt, c, pr, l) =

$$
\begin{array}{lll}
\textit{if} & (t > T_{pmax}) & \\
 & \quad \textit{then} & f_i = 0 \\
\textit{elseif} & (t < T_{pmin}) & \\
 & \quad \textit{then} & f_i = \sqrt{t^{pp_1} * pt^{pp_2} * c^{pp_3} * pr^{pp_4} * l^{pp_5}} \\
\textit{elseif} & (pt > PT_{pmax}) & \\
 & \quad \textit{then} & f_i = \sqrt{t^{pp_1} * pt^{pp_2} * c^{pp_3} * pr^{pp_4} * l^{pp_5}} \\
\textit{elseif} & (pr < PR_{pmin}) & \\
 & \quad \textit{then} & f_i = \sqrt{t^{pp_1} * pt^{pp_2} * c^{pp_3} * pr^{pp_4} * l^{pp_5}} \\
\textit{elseif} & (l < L_{pmin}) & \\
 & \quad \textit{then} & f_i = \sqrt{t^{pp_1} * pt^{pp_2} * c^{pp_3} * pr^{pp_4} * l^{pp_5}} \\
\textit{else} & & f_i = t^{pp_1} * pt^{pp_2} * c^{pp_3} * pr^{pp_4} * l^{pp_5}
\end{array}
$$

where, for the purpose of stating the importance function in a more succinct form, we made the following notations for the (situation dependent) importance space dimensions:

t = $Time_p$, pt = PTime, c = Consequences, pr = PResources, l = Likelihood.

The two kinds of parameters involved are:

- (conditional) preplanning behavior model parameters: pp_1 to pp_5;

- parameters specified by the expert model: T_{pmax}, T_{pmin}, PT_{pmax}, PR_{pmin}, L_{pmin}. They are domain dependent and are defined by the expert specifying the domain knowledge. Their meaning is defined below.

○ the *Expert Model* reflects the new dimensions of the importance space. It must specify the following:

- functions:

  ✧ f_{tc}: transforms (as for reaction decision) real-time values into time-pressure values, inversely proportional, so it has the general form:

$$Time_p = f_{tc}\ (Time_{rc}) = k\ /\ Time_{rc}$$

- parameters:

  ✧ T_{pmax} - time pressure threshold - for greater time pressure, any attempt of response is useless (akin to T_{max} for reaction decision);

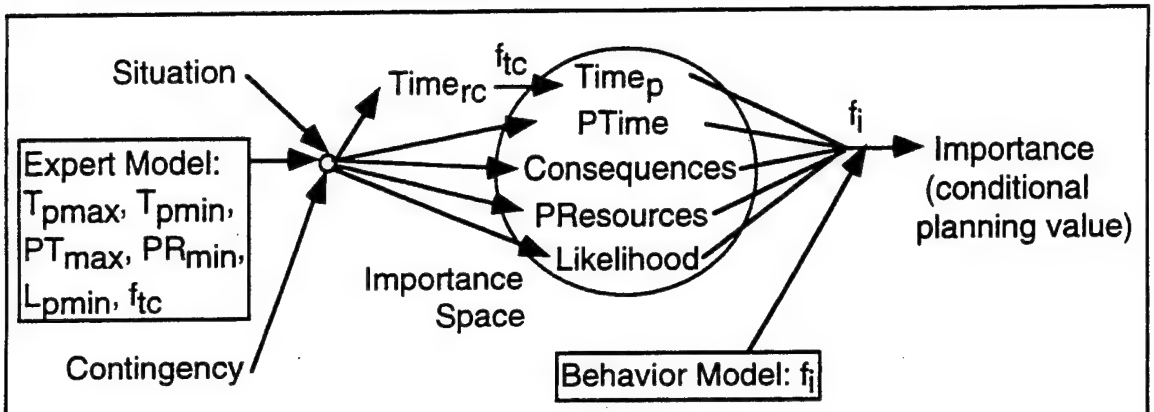
  ✧ T_{pmin} - time pressure threshold - for smaller time pressure, dynamic replanning is possible (and thus less costly, since it will be done only if the contingency actually arises); akin to T_{min} for the reaction framework;

  ✧ PT_{max} - planning time threshold - if the estimated planning time required is longer than this threshold, then the agent may not be able to complete the conditional branch in the estimated available planning time;

  ✧ PR_{min} - replanning resources threshold - for smaller values, the agent has enough execution time resources such that replanning is possible (and presumably less costly);

- ✧ *Lmin* - likelihood threshold - if lower likelihood, the cost of preparing a conditional branch for this contingency in this situation is probably unjustified (same as for the reaction decision framework).

○ the parameters of the *Behavior Model* (pp_1 to pp_5) also reflect the new dimensions of the importance space as well as the new function computing the importance value.



$$Time_{rc} = f_1 \text{ (Situation, Condition)} \qquad Consequences = f_2 \text{ (Situation, Condition)}$$
$$PTime = f_5 \text{ (Situation, Condition)} \qquad Likelihood = f_4 \text{ (Situation, Condition)}$$
$$PResources = f_6 \text{ (Situation, Condition)} \qquad Time_p = f_{tc} (Time_{rc}) = k / Time_{rc}$$
$$Importance = f_i (Time_p, PTime, Consequences, PResources, Likelihood)$$

Figure 3.19. Establishing the Conditional Planning Value

Note that the time to preplan a conditional branch may be different from the time to replan it at execution time, because of different resources availability and different information availability; in the driving example, when building the plan at home we may have all the necessary maps, some of which may be unavailable when replanning later on during the execution of the initial plan, and obtaining them may be time consuming, thus making the initial planning time shorter than replanning time. On the other hand, when replanning, the agent may have access to more accurate state information than at initial planning time, and therefore the initial planning time may in this case be longer than the replanning time (for example, when the agent must replan its route due to a traffic jam, it has more knowledge about which

alternatives are available for faster traffic flow, than it could have before it actually reached this point in the plan execution).

Also note that side-effects are not taken into account in this framework, since once prepared, the conditional branch is executed as a regular plan which under normal circumstances leads to the final goal (the side-effects were a measure of the risk of not being able to reach the final goal anymore, once the reaction is executed).

3.5.4. Deciding Whether to Plan a Conditional Branch

Figure 3.20 presents the part of the framework concerned with the final decision of whether to prepare a conditional branch for the contingency in the given situation. It is similar to figure 3.11 which shows the reaction decision making phase of the previous framework. We shall outline here the differences between the two frameworks at this stage:

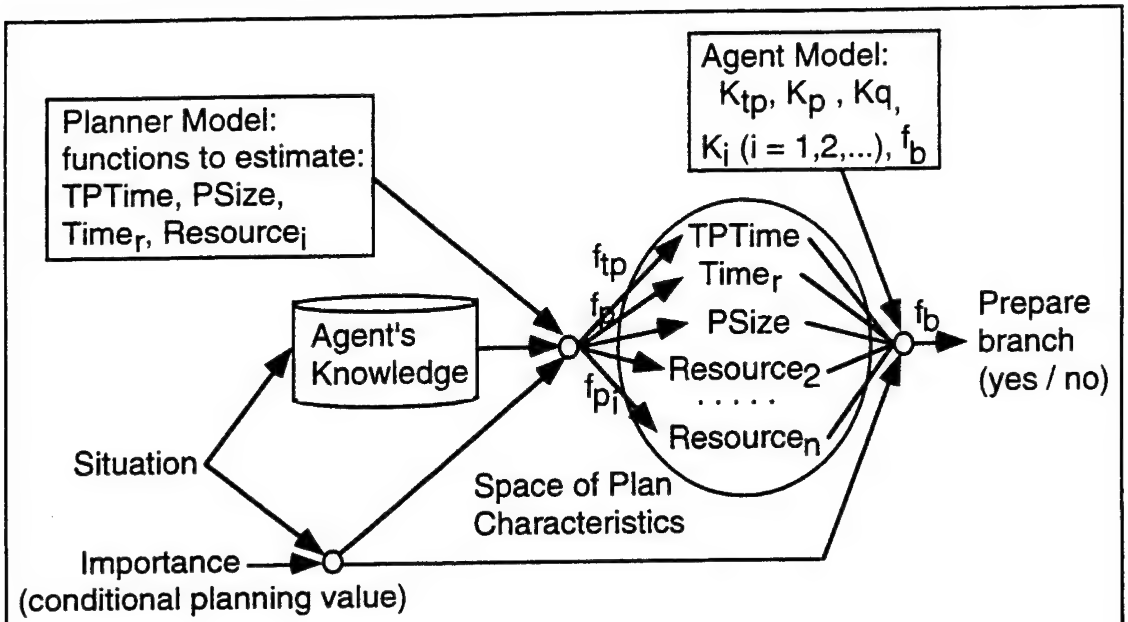
- the reactive plan characteristics space is replaced by a *Plan Characteristics Space* whose dimensions characterize the entire conditional plan to be built, from the point of view of the agent's planning and execution resources. These dimensions are:

  - *TPTime* - measures the total planning time needed by the planner, if a conditional branch for this contingency will be planned in addition to the main plan and conditional branches for the contingencies already selected for conditional planning;

  - $Time_r$ - is the estimated time needed by the agent to respond, at execution time, to this contingency, given that the conditional plan includes a branch for it together with branches for the contingencies already selected for conditional planning (similar to the reaction framework);

  - $Resource_i$ (i = 1,2,...) - represents the total requirements imposed on the agent's i-th resource by the conditional plan containing a branch for this contingency as well as branches for the contingencies already selected for conditional planning (similar to the reaction framework); an example of such a resource may be

memory amount required by the plan, which is separately represented in figure 3.20 by the total plan size (*PSize*).



TPTime = f_{tp} (Situation, Importance, Agent's_knowledge, Planner_model)

$Time_r$ = f_p (Situation, Importance, Agent's_knowledge, Planner_model)

PSize = f_{p_1} (Situation, Importance, Agent's_knowledge, Planner_model)

$Resource_i$ = f_{p_i} (Situation, Importance, Agent's_knowledge, Planner_model)

Prepare_branch = f_b (TPTime, $Time_r$, PSize, $Resource_2$,..., $Resource_n$,
Agent_model, Importance, Situation)

Figure 3.20. The Conditional Planning Decision Making Phase

- the *Planner Model* reflects the new dimensions of the plan characteristics space. It must supply the following functions to estimate values for these dimensions:
  - f_{tp} - estimates the time needed to build the plan, including a branch for this contingency (in its simplest form, it may simply add the already estimated times to build each individual branch);
  - f_p - estimates the time needed to respond to the contingency when the plan includes conditional branches for it and for all contingencies with higher importance (similar to the reaction decision framework);

- f_{p_i} (i = 1,2,...) - estimates the amount of $resource_i$ needed to respond to the contingency when the plan contains conditional branches for it and for all contingencies with higher importance (similar to the reaction decision framework); for i = 1, the function estimates the amount of memory the agent needs in order to accommodate this conditional plan.

○ the *Agent Model* also reflects the new dimensions of the plan characteristics space. It must specify the following:

- estimated maximum resource amounts that may be allocated by the agent to this task:
  - ✧ K_{tp} - the maximum planning time allowed to build the conditional plan (i.e. before any execution begins)
  - ✧ K_1, K_2, ... - the maximum amount of $resource_i$ (i = 1,2,...) available at execution time (i = 1 for memory availability or, equivalently, plan size)

- functions to estimate resource utilization:
  - ✧ f_{b_p} - the increase in planning time due to the agent's computational overhead at the time of planning; it may be of the form:

$$f_{b_p}(TPtime) = TPtime * K_p$$

where K_p is a factor greater than 1, or:

$$f_{b_p}(TPtime) = TPtime + K_q$$

if the agent can devote itself to planning for this contingency only after some constant time K_q, and so on.

  - ✧ f_{b_i} (i = 0,1,...) - the variation of the availability, at execution time, of $resource_i$ (i=0 for computational time; i = 1 for memory or plan size) due to the fact that the agent cannot devote its entire $resource_i$ exclusively to responding to that contingency (same as

the functions f_{r_i} for the reactive plan characteristics space in the reaction decision framework).

- the function (f_b) making the actual decision for a conditional branch preparation:

$$\text{Preplan} = f_b\,(\text{TPTime}, \text{Time}_r, \text{PSize}, \text{Resource}_2, \ldots, \text{Resource}_n, \text{Agent\_model}, \text{Importance}, \text{Situation}) =$$

$$\left|\begin{array}{llll}
\textit{if} & f_{b_p}(\text{TPTime}) > K_{tp} & \textit{then} & f_r = \text{no} \\
\textit{elseif} & f_{b_0}(\text{Time}_r) > \text{Time}_{rc} & \textit{then} & f_r = \text{no} \\
\textit{elseif} & f_{b_1}(\text{PSize}) > K_1 & \textit{then} & f_r = \text{no} \\
\textit{elseif} & f_{b_2}(\text{Resource}_2) > K_2 & \textit{then} & f_r = \text{no} \\
\ldots\ldots & & & \\
\textit{elseif} & f_{b_n}(\text{Resource}_n) > K_n & \textit{then} & f_r = \text{no} \\
\textit{else} & & & f_r = \text{yes}\,.
\end{array}\right.$$

$$= \prod_{i=p;0}^{n} (f_{b_i}(\text{Resource}_i) \leq K_i) \;,$$

where resource_p is the planning time, resource_0 is the execution real computational time, and $K_0 = \text{Time}_{rc}$ is the real response time allowed by the contingency for the response to be started without consequences (the time pressure dimension of the importance space values for this contingency).

Figure 3.21 shows a detailed summary of the framework for selecting the contingencies for which complete conditional branches are to be prepared. We shall not continue the discussion on this topic, since this thesis is mainly concerned with developing the reaction decision framework, and we have included the presentation of the conditional planning framework only to point out that, after we have one of the two frameworks well defined and experimentally proved adequate, the other one can be developed using a certain degree of analogy.

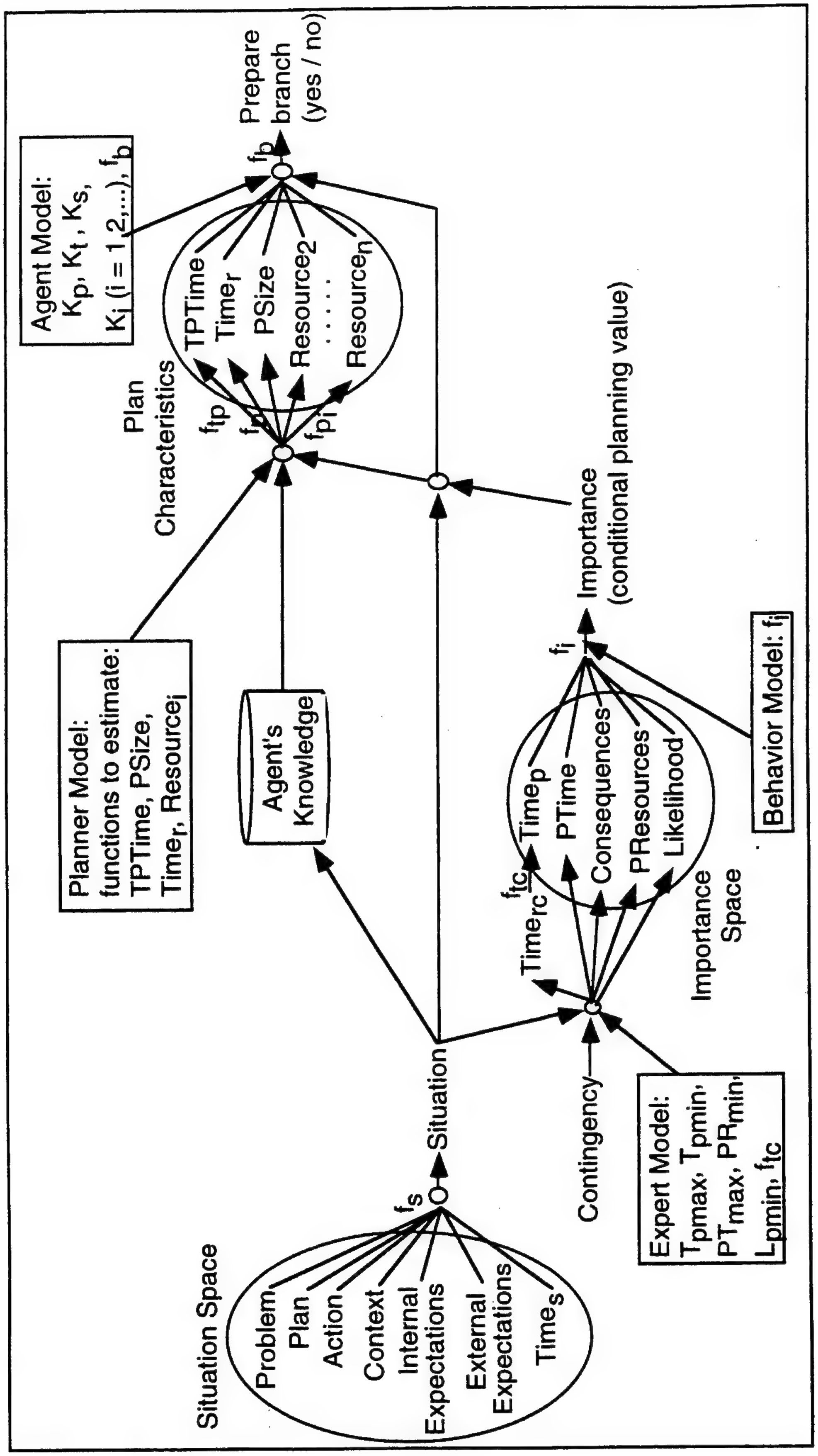


Figure 3.21. The Conditional Planning Decision Framework

Figure 3.22 presents two examples of contingencies that may warrant conditional planning of branches to solve them. They are both taken from the driving domain, but may appear in significantly different circumstances, and they both largely illustrate the way the framework is intended to be applied.

Space	Dimensions	Car driving to work	Car driving to Reno
Situation	Problem	Go from home to work	Go from Palo Alto to Reno
	Plan	Drive car	Drive car on I80
	Context	Morning, commute time	Winter, night time
	Action	Approach intersection B	Approach Sacramento
	Int Expect	Observe traffic light	See Sacramento
	Ext Expect	Heavy traffic	Dark (night time)
	Time	max. 3 mins.	30 mins.
Contingency		Red traffic light (slow - all following lights red too)	Cold & raining hard - maybe snow in mountains
Importance	$Time_P$	To reach intersection B	To reach junction I80, I50
	PTime	High (≈1/2 of main plan)	High (≈1/2 of main plan)
	Consequence	Late for imp. meeting	big delay, maybe life threat
	PResources	Need maps + planning	Need maps + planning
	Likelihood	High (≤ 50% of time)	High
Conditional plan branch		Right turn at traffic light, then alternate route	Use I50 - longer but more reliable when snowing

Figure 3.22. Conditional planning examples

The first example is the one we mentioned in this section before: on the usual commute to work, there is a certain traffic light which, if red on arrival, means that all the following traffic lights will be red, and the commute will take significantly longer than if an alternate route is followed by making a right turn. However, this alternate route is slower if the traffic light in question is found on green.

The second example is set during a trip from the San Francisco Bay area to Reno at night time during winter. If it is cold and raining around Sacramento, then there is a good chance that the usual (and faster) freeway

may be closed in the mountains due to snow, so an alternate route is wiser, but it has to be prepared in advance since it may require maps for planning.

A comparison between figures 3.21 and 3.7 shows that the two frameworks are close enough so that an aesthetically concerned reader can easily merge them into a single framework, so we shall not concern ourselves with this topic anymore. Instead, in the next chapter, we present a knowledge representation formalism to help the agent to cope with the considerable amount of knowledge related to these decision processes.

Chapter 4
Knowledge Representation Formalism

In order to operate in an environment, the agent has to possess a lot of knowledge about that environment. For the purpose of deciding whether to plan to react to possible contingencies according to the framework presented in the previous chapter, the agent has to possess three types of information: knowledge about situations that may be encountered during plan executions, knowledge about the contingencies that may happen in these situations, and knowledge about the most suitable reactions to these contingencies. The agent's knowledge base contains associations of contingencies and their appropriate reactions. Each pair contingency-reaction is indexed in the knowledge base by the characteristics of the situation in which the contingency may appear and in which that is the most suitable reaction to it. Therefore, each condition stored in the knowledge base has three parts:

(i) a description of the contingency (signs, preconditions, and so on) and a set of values for the dimensions of the criticality space

(ii) a description of the best suited reaction for this contingency in the situation described by the third part

(iii) a description of the situation in which this contingency may appear and in which the best response to it is the reaction described in part (ii). This description contains the values for each of the seven dimensions of the situation space mentioned in chapter 3.

In the previous chapter we have presented the kind of information associated with each of these classes of knowledge. With the exception of the contingency information which contains numerical values for the values of the characteristics of the criticality space, the rest of the information is symbolic. This includes the values for the situation space dimensions, the descriptions of the contingencies, and the descriptions of the actions which make up the reactions to contingencies. Theoretically, one could use the natural language to specify these values. However, such a natural language interface and the mechanisms to process the information in such a formalism are beyond the scope of this work. In order to contain the explosion in complexity generated by such a natural language representation, we have defined a knowledge representation formalism which restricts the description language for each of the classes of knowledge under consideration, while retaining enough flexibility to be suitable to any domain and with the added advantage of a well defined structure which can be used in the reasoning process.

In this chapter we shall discuss this knowledge representation formalism for each of the classes of knowledge involved, with examples from the driving domain. We shall first present the general idea which is applied to all the three classes, and then we shall discuss an example of representing the contingency description knowledge for the car driving domain. Appendix 2 presents an example of representations of reactions and representations of situations for the same domain.

4.1. Description Languages

The need to devise a knowledge representation formalism for describing situations, contingencies and reactions has arisen from two considerations:

(i) the space of all possible natural language descriptions for these classes of knowledge is too large to be manageable; this in turn generated problems like the possibility of having different representations for the same piece of knowledge and the associated difficulty of comparing such representations and deciding on their identity. For example, in the car driving set of reactions we have used during the previous chapter,

"steer" may be equivalent to "change direction", and clearly each situation has many different equivalent ways of being described.

(ii) the practical application domains for the framework of deciding whether to prepare to react presented before have a significant amount of inherent structure implicitly contained in them and it would be unfortunate not to be able to exploit this structure. Notice for example that eleven out of the thirteen examples of contingencies we gave for the car driving domain (table 3.1) use the action "brake" in the description of their associated reactions. The car driving domain has also a significant amount of inherent structure in the description of the possible contingencies. For example, the following two contingencies: "Child runs from right, 20 m in front of car" and "Adult crosses the street from right 20 m in front of car" have both the same criticality space values, and the same associated reactions, and therefore do not need separate representations in the agent's knowledge base.

If the structure of the application domain is not taken into account, the explosion of the information that has to be recorded in the agent's knowledge base quickly exceeds any realistically manageable amount for agents operating in the real-world domains described in chapter 2. For example, there are any number of individual situations for which the same pair contingency-response applies, and it would be entirely unreasonable to represent each of them and all their associations with different conditions.

Given these considerations, we have designed a representation formalism for these classes of knowledge which preserves most of the flexibility of the natural language representation, while allowing the expert to take advantage of the structure of the domain.

For each domain there are nine languages which must be defined: a language for describing the contingencies, one for describing the reactions, and seven languages for describing the values associated with each of the seven situation space dimensions. Each of these languages will be described according to the same formalism, so we shall only describe the formalism once, and then (in the following section) we will give an example of each such language in the driving domain.

The expert is required to define a hierarchical vocabulary for each of these languages in his domain. The words in the vocabulary are partitioned into two classes: terminals and nonterminals. Each nonterminal represents a class of words (both nonterminals and terminals). The terminals are classes by themselves. The expert must also define all the membership and subclass relationships among the words in the vocabulary. Each such relationship defines a directed edge in a tree (actually a forest, since there is no need for full connectionism in a vocabulary) which induces a hierarchy onto the vocabulary. The tree is actually an AND/OR graph, in which the OR nodes represent the membership and subclass relationships, and the AND nodes represent structural relationships among words in a valid sentence in the language. Our formalism has a few common features with the language representation formalism presented in [Utgoff, 1988], although it differs in many other aspects. Our formalism defines a context free grammar:

$$G = (N, T, P, S), \qquad \text{where}$$

- N - is the set of nonterminal words of the domain dependent vocabulary defined by the expert
- T - is the set of terminals in the vocabulary
- P - is the set of productions of the grammar; there are two types of productions:
  - unit productions, defined by a membership relation between a terminal and a nonterminal or by a subclass relation among two nonterminals
  - non-unit productions, defined by AND nodes in the vocabulary - these give the rules of correct derivations in the language.
- S - the start symbol, which is either the root of the tree (if one exists), or, if the vocabulary is organized as a forest, then it is a new nonterminal (OR node) to which all the roots of the trees making up the forest are connected through subset relationships edges.

This context free grammar defines the language used for describing either the contingencies in the domain, or the reactions, or one of the seven

characteristics of the situation space, with one important difference from the classic theory of context free languages: every word in the vocabulary may be part of a sentential form in the language, that is, both terminals and nonterminals may be used to build sentential forms. The set of terminals in the vocabulary makes up the agent language, that is, the set of all individual describable contingencies (or reactions, or characteristics of situations). A sentential form containing only terminals represents a description of a specific contingency, reaction, or situation characteristic. It can also be interpreted as a description of a singleton set of contingencies, reactions or situation characteristics. A sentential form containing at least a nonterminal symbol represents a description of a set (of any cardinality) of such contingencies, reactions or situation characteristics. This extension of the context free grammar paradigm enables us to represent the structure of the application domain.

Our formalism also extends the classical context free grammar paradigm with the notion of identification functions for nonterminals in the vocabulary. An identification function is a compact way of representing a large set of class membership relationships or a large set of subclass relationships. For example, the nonterminal "slow_driving_speed" can be identified by a function defined as:

$$f\,(\text{speed}) = 5 \text{ mph} < \text{speed} < 20 \text{ mph}.$$

This function replaces all the edges in the tree between the nonterminal "slow_driving_speed" and all the terminals "speed = x" where x can take all the discrete values representable in the machine (or in the defined domain vocabulary) between 5 mph and 20 mph.

Every tree generated by a vocabulary as described above defines two partial order relations among the words of the vocabulary as well as among the set of sentential forms that can be built. The elementary partial order relation, which we call "*contains*", among words in the vocabulary, is defined as: "*a* contains *b*" if and only if *a* and *b* are words in the vocabulary, *a* is a nonterminal, and either *a* and *b* are identical, or *a* contains *b* as a member (if *b* is a terminal), or *a* includes *b* as a subset (if *b* is a nonterminal). The extended partial order relation with the same name is applied to sentential forms

through the following definition: "A contains B" if and only if A and B are sentential forms in the language (according to the previous definition), and every word in A contains the word in B in the corresponding position, i.e.

$$\text{if} \quad A = a_1 a_2 \ldots a_k \quad \text{and} \quad B = b_1 b_2 \ldots b_n$$

then a_1 contains $b_1 b_2 \ldots b_{p_1}$, a_2 contains $b_{p_1+1} b_{p_1+2} \ldots b_{p_2}$, and so on until a_k which contains $b_{p_{k-1}+1} b_{p_{k-1}+2} \ldots b_n$.

In the next section we shall give an example of applying the formalism described here to the car driving domain and we shall present the vocabulary trees which can be used to express the contingencies given in table 3.1. The effectiveness of this representation formalism in structuring the application domain will be illustrated by the realization that the same vocabulary tree allows for the representation of a much larger set of contingencies, with essentially the same knowledge acquisition effort and similar storage and computational requirements. We shall then conclude this chapter with a brief summary of the advantages of this knowledge representation formalism.

4.2. Example

In this section we shall present the hierarchical vocabulary (and consequently the grammar) which are sufficient to represent the thirteen contingencies for the car driving domain listed in table 3.1. Appendix 2 contains a description of the vocabulary for representing the reactions, and those for the situations encountered in chapter 3. The vocabularies will not only be able represent the knowledge contained in table 3.1, but also a lot more.

Figure 4.1 presents the hierarchical vocabulary for representing contingencies.

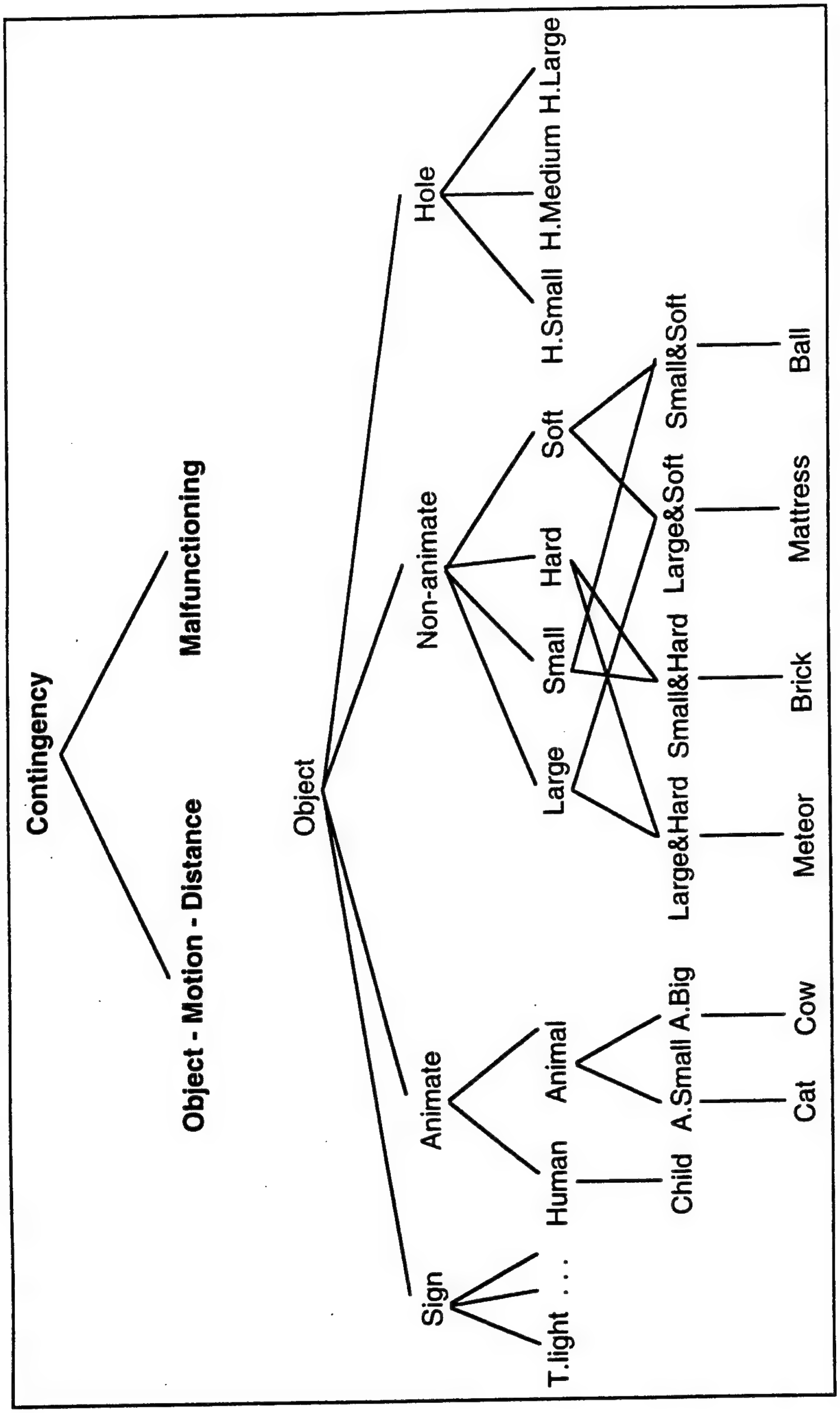


Figure 4.1. Vocabulary for describing contingencies in the driving domain

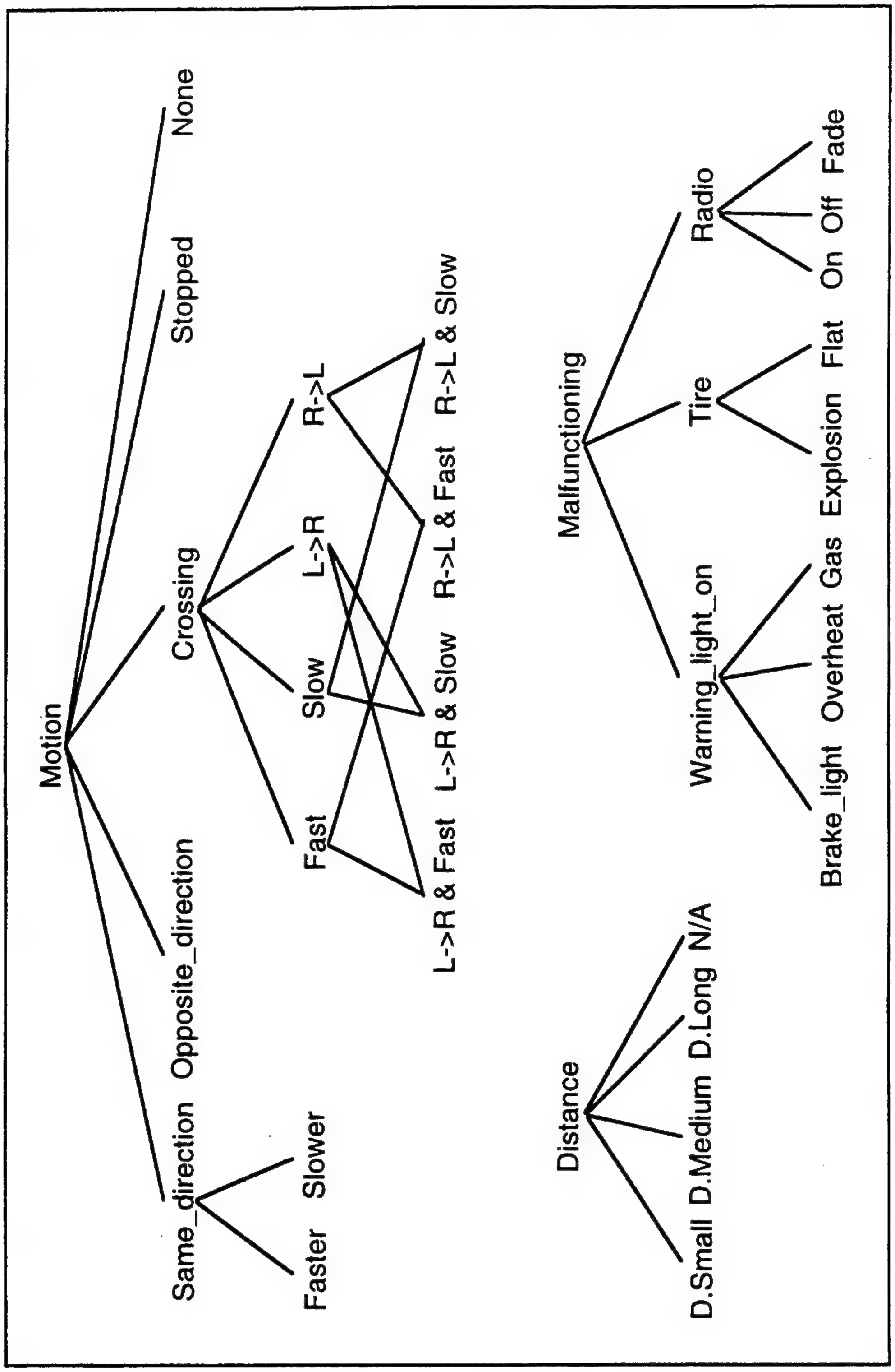


Figure 4.1. Vocabulary for describing contingencies in the driving domain (continued)

This hierarchy is equivalent to the following grammar:

G = (N, T, P, S), where:

N = { Contingency, Object, Motion, Distance, Malfunctioning, Sign, Animate, Non-animate, Hole, Human, Animal, Large, Small, Hard, Soft, A.Small, A.Big, Large&Hard, Small&hard, Large&Soft, Small&Soft, Same_direction, Crossing, Fast, Slow, L->R, R->L, Warning_light_on, Tire, Radio }

T = { T.light, Child, Cat, Cow, Meteor, Brick, Mattress, Ball, H.Small, H.Medium, H.Large, Faster, Slower, Opposite_direction, L->R&Fast, L->R&Slow, R->L&Fast, R->L&Slow, Stopped, D.Small, D.Medium, D.Long, Brake, Overheat, Gas, Explosion, Flat, On, Off, Fade }

P = { Contingency -> Object - Motion - Distance | Malfunctioning
Object -> Sign | Animate | Non-animate | Hole
Sign -> T.light | . . .
Animate -> Human | Animal
Non- -> Large | Small | Hard | Soft
Hole -> H.Small | H.Medium | H.Large
Human -> Child | ...
Animal -> A.Small | A.Big
Large -> Large&Hard | Large&Soft
Small -> Small&Har | Small&Soft
Hard -> Large&Hard | Small&Hard
Soft -> Large&Soft | Small&Soft
A.Small -> Cat | . . .
A.Big -> Cow | . . .
Large&Hard -> Meteor | . . .
Small&Hard -> Brick | . . .
Large&Soft -> Mattress | . . .
Small&Soft -> Ball | . . .
Motion -> Same_direction | Opposite_direction | Crossing | Stopped
Same_direction -> Faster | Slower
Crossing -> Fast | Slow | L->R | R->L

Fast -> L->R&Fast | L->R&Slow | R->L&Fast | R->L&Slow
Distance -> D.Small | D.Medium | D.Long | N/A
Malfunctioning ->Warning_light_on | Tire | Radio
Warning_light_on -> Brake_light | Overheat | Gas
Tire -> Explosion | Flat
Radio -> On | Off | Fade }

S = Contingency

Some derivations may be done through identification functions. For example, the grammar symbols D.Small, D.Medium, D.Long can be considered nonterminals (instead of terminals like in the previous example), and the actual values of the distance can be considered terminals. Then, a function like:

D.Small = 5 m < distance < 25 m

can be used to perform the transition over the edge linking D.Small with the actual terminal, say "distance = 20 m".

Every contingency in table 3.1 can now be obtained through a number of different derivations in this grammar, and since the reactions to them usually apply to more general contingencies, the derivation can be stopped at higher levels, since a sentential form can contain both terminals and non-terminals in the grammar. For example, the contingency:

"Child runs from right 20m in front of car"

can be obtained through the following derivation:

Contingency ->
Object - Motion - Distance ->
Animate - Motion - Distance ->
Animate - Crossing - Distance ->
Animate - Crossing - D.Small ->
Human - Crossing - D.Small ->
Human - Fast - D.Small ->
Human - R->L&Fast - D.Small ->
Child - R->L&Fast - D.Small ->

Child - R->L&Fast - distance=20m.

Any sentential form encountered during this derivation (or during any other derivation leading to the same contingency) can be used to denote this contingency. Each such sentential form contains (and denotes) the set of all contingencies derivable from it. The same reaction specified for this contingency in table 3.1. (Brake hard and steer right) would probably be recommended for the entire set of contingencies: "Human - R->L&Fast - D.Small", while the consequences of the contingency would probably have the same value for an even larger set of contingencies: "Human - Crossing - D.Small".

Clearly, this small vocabulary is not enough to describe all possible contingencies in the driving domain. It was not our goal to provide such a vocabulary and grammar. However, while every contingency in table 3.1 can be derived in this formalism, it supports the derivation of many other contingencies for the driving domain. In fact, just by enlarging the set of terminals, the number of contingencies expressible with this small grammar becomes very large indeed. This fact underlines the most important advantage of this representation formalism, namely imposing a (hierarchical) structure on the set of possible contingencies in the domain, which then makes them much easier to be stored, managed, analyzed and reasoned about.

The knowledge representation formalism used in this chapter allows for collapsing entire sets of contingencies in categories, thus alleviating the problem of knowledge base size explosion.

Another advantage of this representation formalism is that it can be used in a future work for learning purposes, that is for learning which sets of contingencies are similar from certain points of view of the general framework for deciding whether to plan to react introduced in this paper: which contingencies have the same characteristics, or the same reactions, or may appear in the same situations. Concept learning mechanisms ([Mitchell, 1978; Mitchell & al., 1983; Dabija, 1990]) can be applied to contingencies represented in this formalism, mainly because the terms "classification rule" and "concept description" used in machine learning are synonyms with "set description", which represents any sentential form derivable in this

formalism. This representation can also be used to discover new classes of contingencies (non-terminals in the vocabulary) which have eluded the expert's attention when specifying the domain, through bias shifting (either automatically [Utgoff, 1988], or interactively with the expert [Dabija & al., 1992a,b]).

The primary disadvantage of the knowledge representation formalism described in this chapter is that the expert must define the structure of the domain, that is, must specify both the nonterminals of the grammar (not just the terminals), and the membership and subclass relations among the elements of the vocabulary. This may place some burden on the expert and may make the knowledge acquisition process more difficult. Another disadvantage is that each specified vocabulary is domain-dependent (and even user-dependent), as are all the relationships expressed through this formalism. They all reflect how the expert who participated in the knowledge acquisition process views the domain. But the advantages (mentioned above) of structuring the domain and significantly reducing the size of the knowledge base outweigh by far this disadvantage, with the added benefit that the expert is himself compelled to structure his own knowledge of the domain. These problems may further be alleviated by using the learning techniques mentioned above: some of them will attempt an automatic restructuring of the knowledge base, while others will help the expert to structure his own knowledge of the domain through interactions with the system. However, no knowledge acquisition work has been done as part of this thesis.

The entire previous discussion applies equally well to representing reactions and situations. Hierarchical vocabularies may be used to classify reactions since in real domains there are usually a small set of actions which can be combined to produce useful reactive plans, which are then associated with classes of (rather than individual) contingencies. This allows a better structuring of the set of reactions, which in turn ensures better analysis and facilitates the reasoning about different sets of related reactions and their characteristics with respect to the framework presented in the previous chapter.

The same is true for representing situations. Here this representation formalism is even more useful since the variety of situations in real domains

in virtually infinite, so any mechanism which induces a certain structure and facilitates the reasoning process is more than welcome. Identification functions are also particularly useful here, since the values of some of the dimensions may belong to continuous sets. Classes of situations defined through this knowledge representation formalism and satisfying the "contains" relation, are used to more efficiently index contingencies and reactions in the knowledge base (as opposed to indexing them to specific situations, which would be prohibitive in any reasonably-sized real domain). The vocabulary for representing situations may be partitioned into seven distinct vocabularies, one for each dimension of the situation space. Alternatively, for uniformity of presentation reasons, we can combine the seven vocabularies into a single one, with a new start symbol "*Situation*", by adding to the grammar a new production of the form:

Situation -> Problem - Plan - Context - Action - Internal_Expectations - External_Expectations - Time,

where *Problem, Plan, Context, Action, Internal_Expectations, Time* and *External_Expectations*, were the start symbols for each of the vocabularies for the seven dimensions of the situation space.

The hierarchical vocabularies (and the grammars they generate) for representing the reactions listed in table 3.1 for the car driving domain, and for representing certain situations in this domain (including the one used throughout chapter 3) are presented in appendix 2. Some derivations of sentential forms encountered in chapter 3 for reactions and situations in the driving domain are also discussed in appendix 2. As in the case of contingencies, these vocabularies can represent a much larger set of reactions and situations than the ones we have encountered during our presentation in the previous chapters, with very little or no overhead. This once again supports our claim regarding the power of the knowledge representation formalism presented here, and outweighs by far its disadvantages.

Chapter 5
Theoretical Analysis

The dream of any designer is to prove that his product is the ideal one to solve the original problem that motivated the design. In our case, this would mean proving that the framework introduced in chapter 3 is always able to decide, for any given situation, which of a set of contingencies possible in that situation should be selected at planning time to prepare reactive responses for. It would also mean to prove that this is the simplest framework with this property, and also that the set of contingencies selected by it make the best possible use of the agent's execution time resources. But since our objective is to design a framework that is applicable in the most demanding real-life domains, theoretically proving all the previous properties is beyond our means. However, we have been able to theoretically justify some of these properties and some weaker versions of others. For the rest, while we do believe that they hold in our case, we could only provide experimental justifications which are presented in the following chapter.

In this chapter we present the theoretical justifications for a few of the properties stated above. We first justify (through counterexamples) our claim that each of the elements included in the framework is necessary, that is that the framework is free of redundancies. Next we claim that the framework can consistently implement desired behavior models, and that the criticality function defined in section 3.3.2 can implement any known type of reactive behavior; we formally justify the first of these claims, and in the next chapter we present an experimental justification for the second one. Finally, we also

claim that the set of contingencies selected through our framework makes the optimal use of the agent's execution time resources while simulating the desired reactive behavior pattern, and we formally justify it. One more claim which cannot be justified theoretically but is verified experimentally in the next chapter is that the knowledge required by our framework in order to execute properly exists and can be acquired in real domains.

But in the next section let us first briefly review the general assumptions of our framework, which will be used during this chapter. In the following sections we shall then present our theoretical justifications to the properties of necessity, consistency and optimality of the framework.

5.1. Assumptions

As discussed in chapter 2, during our presentation we have made certain assumptions about the problem we attempted to solve. These assumptions refer to both the agent, and the environment in which it is designed to work. The assumptions regarding the agent refer both to the agent's execution capabilities, as well as to the design of its different control modes.

The main assumptions for designing our framework were:

- about the agent capabilities:
  - planning (and planning to react)
  - monitoring
  - reacting
  - limited resources (including computational time)
- about the task environment:
  - real-time requirements
  - complex - there exist a large (infinite) number of possible situations
  - complex - there exist a large (maybe infinite) number of possible contingencies in each situation
- about the agent control modes:

- planning is better than reaction, whenever the resources (including execution time) allow it
- planning (like reaction) is useless whenever there is insufficient time to reach a solution
- reaction is faster than planning
- limited resources allow only for limited amounts of reaction

We also assume that the agent's knowledge base always contains whatever information may be necessary for the operation of the framework. Whether such information exists in real life and whether its acquisition by the knowledge engineer or the agent is possible will not be of concern in this chapter. However, we claim that this information indeed exists and its acquisition is not very difficult, and we support our claim with the experiments described in the next chapter and performed in different domains requiring quite different types of human expertise.

Note that all the assumptions listed here are not very restrictive. In fact, they mostly restate the applicability conditions for our framework, presented in chapter 2. This means that the following results do not lose their generality from these assumptions.

Any other local assumptions that we shall make in order to allow us to perform theoretical analyses of our particular claims will be stated whenever they apply.

5.2. Necessity

We claim that each element of our framework is indispensable for the final decision, that is that each element in the framework is necessary for the final decision, or alternatively, that the framework is free of redundancies. The simplest way to justify this claim is to assume that each element of the framework is redundant (one at a time) and then disprove this assumption by presenting a counterexample. This also proves that the elements of the framework are independent (uncorrelated). To do this, we specify a complete decision problem (again in the car driving domain since now we are very familiar with it) and then change the values of each element of the

framework, one at a time, and show that this potentially yields a different decision each time. This implies that if that element of the framework is missing, then an ambiguity is allowed in the decision process.

Property: The framework presented in chapter 3 for deciding whether to plan to react to a given contingency in a given situation is free of redundancies, i.e. each element of the framework is necessary for the final decision.

Justification: we shall state a problem, assume in turn that each element of it is redundant, and show by counterexample that this is not true.

Example problem:

Variables:

Situation Space:

Problem:	Carry book from home to the office
Plan:	Drive car
Context:	School time (Weekday, morning, May)
Action:	drive straight on Street S, 25 mph
Internal Expectations:	reach school
External Expectations:	children in sight
$Time_S$:	1 - 3 minutes

Contingency:	Ball in front of car

Criticality Space:

$Time_P$ ($Time_{rc}$):	very high (very short) (9)
Consequences:	small (3)
Side-effects:	medium-high (6)
Likelihood:	medium (5)

Parameters:

Expert Model:

T_{max} = 9.5; - maximum time pressure allowed to respond

T_{min} = 3; - minimum time pressure required to react

CS_{min} = 4; - maximum difference allowed between side-effects and consequences

L_{min} = 3; - minimum likelihood required to react

MON = 1000; - minimum criticality required to monitor

Agent's Knowledge:

7 contingencies: 4 of higher criticality than this one,
2 of lower criticality than this one

Reactive Planner Model:

decision trees:

f_t: log: 0.2 * $\log_2$(nr_of_conting_with_greater_criticality)

Agent Model:

computational overload - implies computational time delay:

f_{r_0}: 1.3 * $time_r$

Behavior Model: "normal"

Parameters for the criticality function f_c: $p_1 > p_5 > p_6 > p_2$

$p_1 = 5$; $p_2 = 1$; $p_3 = 0$; $p_4 = 0$; $p_5 = 3$; $p_6 = 2$

Changing one element of the framework at a time produces the following changes in the criticality space values and implicitly in the reaction value of this contingency (which imply changes in the order of including the contingencies in the reactive plan):

Situation Space:

Problem: Carry 3 kg of radioactive material
Changes: increases Side-effects

Plan: Ride a bike
Changes: increases Consequences
decreases Side-effects

Context: Night-time (non-school time)
Changes: decreases Likelihood

Action: drive straight, 40 mph
Changes: increases Consequences
increases Side-effects
increases Time pressure

Internal Expectations: reach railway crossing
Changes: decreases Likelihood

External Expectations: train in sight
Changes: decreases Likelihood

$Time_S$: ≤ 0.5 seconds
Changes: decreases Likelihood

Note: Any of the changes in the situation space dimensions mentioned also changes the set of possible contingencies which include the one under consideration. Some of the changes add contingencies with high criticality, so this contingency will get a smaller priority of being considered for reactive response, others have the opposite effect.

Contingency Child in front of car
Changes: increases Consequences

Expert Model

T_{max}: lower (8.5)
Changes: decreases Criticality (as a whole) since $time_p$ (9) becomes greater than T_{max} (8.5)

T_{min}: higher (9.1)
Changes: decreases Criticality (as a whole) since $time_p$ (9) becomes smaller than T_{min} (9.1)

CS_{min}: lower (2.5)
Changes: decreases Criticality (as a whole) since the difference side_effects - consequences (3) becomes greater than CS_{min} (2.5)

L_{min}: higher (6)

Changes: decreases Criticality (as a whole) since likelihood (5) becomes smaller than L_{min} (6)

MON: higher (i.e. higher than the criticality of this contingency)

Changes: do not even monitor (or prepare to react to) this contingency

Agent's Knowledge:

larger: 24 critical contingencies (more critical than this one)

Changes: the chances to prepare reaction to this contingency decrease because it has a low reaction value compared to the other contingencies known for the same situation

Reactive Planner Model:

decision lists:

f_t = linear: 0.2 * nr_contingencies_with_greater_criticality

Changes: increases real response time

Agent Model:

f_{r0}: 1.8 * $time_r$

Changes: increases real response time which may determine it to exceed $time_{rc}$ and thus to be excluded from the reactive plan

Behavior Model: - changes in the criticality function's (f_c) parameters:

p1: lower (1)

Changes: decreases criticality - disregards allowed response time

p2: higher (3)

Changes: increases criticality - stresses consequences

p3: higher (2)

Changes: increases criticality - stresses side-effects

p4: higher (2)

Changes: increases criticality - stresses anything that can go wrong (both consequences and side-effects)

p5: lower (1)

Changes: decreases criticality - disregards consequences

p6: higher (5)

Changes: increases criticality - stresses likelihood (prepares first for the most frequent contingencies)

All these changes in the parameters values of the criticality function denote a change in the behavior model implemented by the framework, and have as effect a change in the ordering of contingencies by reaction value, which may yield a different set of contingencies to be selected for reactive response.

❑

This concludes our justification that each element of our framework is necessary for the final decision, or alternatively, that the framework is free of redundancies. We have shown that for any such element, there may be a variation in its value which may determine a different outcome of the final decision, and also that such a variation in this value is possible (and even plausible) in the domains under consideration.

5.3. Consistency

I would have liked to be able to say that I proved that the framework introduced in chapter 3 is always able to decide, for any given situation, which of a set of contingencies possible in that situation should be selected at planning time to prepare reactive responses for. This would obviously solve this problem forever, and we could all do something else. But since our objective is to design a framework that is applicable in the most demanding

real-life domains, theoretically proving this property is beyond our means. However, we are able to theoretically justify a few weaker properties which would still ensure the usefulness of the framework. On an encouraging note, the previous claim actually held in the domains in which experiments were conducted. And since these domains are significantly varied in nature, we may still conclude that it will be true for a large set of real-world domains.

We present here the theoretical justification for our claim that the framework for deciding whether to plan to react defined before can consistently implement behavior models. This actually means that the order in which the contingencies associated with a certain situation are classified according to their reaction value by our framework is the same order as given by the behavior pattern under consideration.

In order to construct our justification, we start with a few preparatory definitions and we will prove a few other properties along the way too.

Definition: An *Evaluation function* (f_e) is a function which, given a set of conditions (pairs contingency-reaction) and a situation in which they apply, computes a score, with the property: the higher the score, the better (more appropriate) that set of contingencies is, according to a particular reaction philosophy (behavior model).

Definition: A *Behavior model* is an order relationship on the set of contingencies associated with a situation.

The behavior model represents the type of reactive behavior exhibited by the agent, that is, given any pair of contingencies and their reactions in a situation, which contingency is to be preferred by the agent for reaction (i.e. has priority in reacting to, and hence in preparing a reaction for).

Obs.: there is a functional relationship between evaluation functions and behavior models, i.e. every evaluation function characterizes a behavior model, but a behavior model may be characterized by a set of evaluation functions.

Definition: A *Rational behavior* is a subset of conditions (pairs contingency-reaction) such that, given an evaluation function and

an agent with limited resources, there is no other subset of conditions that gives a better score for this function while satisfying the resource limitations.

The notion of rational behavior has been defined independently of the situation characteristics, because all the contingencies that belong to the same subset must first of all apply to the same situation. The only contribution of the situation space to the framework is to uniquely define each situation, and thus unambiguously identify the contingencies and the reactions associated with it.

The criticality function f_C (section 3.3.2) defines an order relation, called "*more important* " on a set of conditions matching a given situation.

Definition: Condition a is *more important* than b *in situation* S ($a >_S b$) if and only if:

(i) both conditions a and b match situation S

(ii) in situation S: $f_C(a) > f_C(b)$, i.e. the criticality value of a is higher than that of b.

Obs: "more important" is not a partial order relation on the entire set of conditions in the agent's knowledge base, because there may be two situations (S and T) in which both contingencies *a* and *b* may appear and such that $a >_S b$, and $b >_T a$. Therefore, the relation "more important" is only defined in a given situation.

Property: The sum of the criticality values (reaction values) for a set of conditions is an evaluation function.

Justification: Let $f_C(c)$ be the reaction value of condition (pair contingency-reaction) *c* in situation *S*, and let *C* be a set of conditions associated with situation *S*. Then:

$$f_e(C) = \sum_{c \in C} f_C(c)$$

is an evaluation function. Indeed, according to the previous definition of an evaluation function, f_C computes a score for a set of

conditions in a situation, and since $f_c(c) \geq 0$ for any $c \in C$ (according to its definition in section 3.3.2), f_e can characterize a behavior model. This is true because for any condition *a* and any set of different conditions *C*, $C \cup \{a\}$ must be preferred to C by the behavior model (it is never worse to be prepared to react to more contingencies, when agent resource limitations are not taken into account, and here the behavior model has been defined independently of the agent's resource limitations).

❑

Property: For any two conditions *a* and *b* associated with a same situation *S*, we have: $a >_S b$ if and only if, in situation *S*, the behavior model prefers condition *a* to condition *b*, i.e. it requires the agent to attempt to include the reactive response for condition *a* before attempting to include a reactive response for *b* (i.e., according to the behavior model, given a choice, it is more important that the agent is prepared to react to contingency *a* than to contingency *b*).

Justification: If $a >_S b$ then $f_c(a) > f_c(b)$ in situation *S*, so for any set of conditions *C* not including *a* and *b*:

$$f_e(C \cup \{a\}) = \sum_{c \in C} f_c(c) + f_c(a) > \sum_{c \in C} f_c(c) + f_c(b) = f_e(C \cup \{b\}),$$

so the evaluation function gives a higher value for $C \cup \{a\}$ than for $C \cup \{b\}$, and thus the behavior model requires the agent to attempt to include *a* before *b* in the reactive plan associated with situation *S*.

Conversely, if the behavior model requires the agent to attempt to include *a* before *b* in the reactive plan associated with situation *S*, then the evaluation function for this behavior model gives a higher value for $C \cup \{a\}$ than for $C \cup \{b\}$, for any set of conditions *C* applicable in situation *S* and which do not include *a* and *b*, i.e.: $f_e(C \cup \{a\}) > f_e(C \cup \{b\})$, i.e.:

$$\sum_{c \in C \cup \{A\}} f_c(c) > \sum_{c \in C \cup \{B\}} f_c(c), \qquad \text{that is:}$$

$$\sum_{c \in C} f_C(c) + f_C(a) > \sum_{c \in C} f_C(c) + f_C(b),$$

and so $f_C(a) > f_C(b)$ in situation S, i.e. $a >_S b$.

❑

Property: The framework presented in chapter 3 for deciding whether to plan to react to a given contingency in a given situation consistently implements behavior models.

Justification: The notion of behavior model is only about the preferences of reacting to contingencies, and thus it is only connected to the notion of reaction value, implemented in the framework by the criticality function. The previous property shows that the "more important" relation introduced by the criticality function orders the contingencies applicable in a situation in the same way as the preferences of the behavior model described by this criticality function. Therefore, the criticality function represents an appropriate way to describe a reactive behavior model in our framework.

❑

Moreover, because of the optimality property proved in the next section, the framework, using the criticality function ordering of contingencies, can always optimally implement the behavior model as restricted by the agent's resource limitations, i.e. the rational behavior.

This concludes our justification for the consistency property of our framework. This last property has stopped short of claiming that our framework is sufficient to simulate any behavior pattern desired, since theoretically there are an uncountable number of behavior models and only a countable number of implementable criticality functions (as a subset of the set of all programs written in a given programming language), so this would have been impossible to prove (actually we just explained it to be false). However, we shall state a much more practical conjecture here.

Conjecture: for each known (cited in the literature) type of behavior, there exists a combination of parameters in our criticality function which implement it.

This conjecture cannot be actually proved, but can be experimentally supported, as discussed in section 6.3. Coupled with the previous property, it implies that the framework yields the rational behavior for the agent given an evaluation function (a behavior model), for any distributions of the set of characteristics for the conditions (including any distribution of deadlines for the reactions to contingencies) and for any distributions of the agent's resources.

If we are unable to come up with a suitable criticality function for a desired behavior model, then any of a significant number of automatic or interactive learning methods may be employed to learn this function, as suggested in chapter 7.

5.4. Optimality

We also claim about our framework that it makes the best use of the execution time resources of the agent. This means that, given a set of contingencies for a situation, the framework will choose not only those contingencies that are most important to be treated reactively (according to its reactive model), but will also select as many as it can so that the reactive plan built for these contingencies maximizes the use of the agent's runtime resources.

We first restate here the definition for a rational behavior introduced in the previous section:

Definition: A *Rational behavior* is a subset of conditions (pairs contingency-reaction) such that, given an evaluation function and an agent with limited resources, there is no other subset of conditions that gives a better score for this function while satisfying the resource limitations.

According to this definition, a rational behavior maximizes the evaluation function for a given situation and agent model, while in the same time producing a behavior pattern consistent with the agent's behavior model.

Property: In the assumptions of section 5.1, an agent, enhanced with the framework presented in chapter 3, exhibits the rational behavior: it maximizes the use of its resources, while simulating the desired reactive behavior pattern.

Justification: The criticality value establishes an order on the set of contingencies associated with the situation, according to the desired evaluation function (conf. section 5.3). The decision process (function f_r in section 3.4.3) is then applied to each of these contingencies, in the order established. There are two possible outcomes of this process for a contingency which was already considered worth monitoring: if there are enough resources (as estimated by the agent model) then the contingency will be included in the reactive plan; otherwise, this contingency will not be included for reactive response. However, this does not mean that the agent's resources were exhausted by the set of contingencies already considered. It only means that the resources left available are not sufficient to respond to this contingency (while still responding in useful time to the ones with higher criticality, already included in the reactive plan). Therefore, our framework continues the evaluation of the remaining contingencies (also in the order of their criticality values) since a less critical contingency may require less resources and therefore can also be included for reactive response. No contingency can be added to this set when each of the remaining contingencies requires more resources than left available by the ones already in the set. Therefore, we conclude that our framework makes the best use of the agent's resources (as estimated by the agent model) given a certain evaluation function (which expresses a specific desired reactive behavior of the agent).

❑

We have thus theoretically justified our claim that the framework we have introduced in chapter 3 for deciding whether to plan to react to a given

contingency in a certain situation yields the rational behavior for the agent given an evaluation function (a behavior model), for any distributions of the set of characteristics for the conditions (including any distribution of deadlines for the reactions to contingencies) and for any distributions of the agent's resources. This fact takes off some burden of our experiments, since we will only have to conduct experiments for the claims which have no theoretical justification. However, we also present, in chapter 6, the results of an experimental demonstration of the rational behavior claim as well as the claims justified in the previous section.

Chapter 6
Experiments

In order to demonstrate the applicability and scalability of the reaction decision framework presented in chapter 3, we have run a number of experiments. We describe here these experiments and the main conclusions that can be drawn from them. In order to demonstrate the generality of the framework, we have conducted the experiments in three different domains: the driving domain from which we took most of the examples used during the previous presentations, and two medical domains of expertise: anesthesiology and intensive care patient monitoring. It is well known that different experts in a domain may have different opinions on specific subjects from the domain. In order to obtain a consensus of these opinions in the driving domain, we have polled 8 experts, and we have combined their opinions in different ways. It was interesting to find out that the results of these combinations had a high degree of similarity among them, and were well in line with the individual opinions of the experts (although among them opinions may have varied significantly). For the medical domains we have only used the advice of a single expert in the field. In the following section we describe the knowledge acquisition process which we have conducted in the driving domain, and its results. Then we describe a set of experiments in this domain, that support the claim of optimality for our framework which has been theoretically justified in the previous chapter. In the third section we present a set of experiments which were aimed to demonstrate how different behavior models can be described in our framework and how they affect the reactive behavior of an

agent using them. We conclude this chapter with a description of how the reaction decision framework proposed here can be included in a complex agent which runs in a real, complex world.

6.1. The Driving Domain

In this domain we were able to collect knowledge from 8 experts, since most people can be considered experts in this domain, and seven of my colleagues (David Ash, Alex Brousilovski, Lee Brownston, Janet Murdock, Serdar Uckun, Rich Washington and Michael Wolverton) were kind enough to volunteer their valuable time and experience for this part of the project. Beside providing the raw knowledge, they have also made significant comments which have helped me clarify the knowledge acquisition problems involved. I am indebted to all of them (the eighth person in the experiment was myself).

	Contingency	Reaction
1	Child runs from right, 20 m in front of car	Brake hard and steer right
2	Car crosses w/o priority 20 m in front, from right to left	Brake and gently steer right
3	Car in front stops suddenly	Brake hard
4	Cat runs across street, 20 m in front	Brake hard and steer right gently
5	Traffic light changes red 40 m in front	Brake hard
6	Tire explosion	Brake gently and do not steer
7	A deep and medium width hole detected 30 m in front	Brake hard and steer right gently
8	Airplane lands in front of car	Brake moderately hard
9	Brake malfunction light turns on	Brake gently
10	Engine overheat light turns on	Brake gently to stop the car
11	Loud radio turns on suddenly	Adjust radio volume
12	Meteor falls on the trunk of the car	Accelerate hard
13	A ball pops in the street, from the right, at 20 m in front[1]	Brake hard and steer right

Table 6.1. Contingencies for the car driving domain experiments

[1] We have specifically excluded the conventional driver's wisdom case that a ball popping up in the street is usually followed by a running child.

Table 6.1 lists the 13 contingencies (also listed in table 3.1 and used for illustration purposes throughout the thesis) proposed, together with the reactions for each of them.

The knowledge acquisition problem was to specify a value between 0 and 10 for three of the criticality space dimensions (consequences, side effects and likelihood), and a real time value for the time to respond to the contingency, for each of these 13 contingencies, when considered possible to appear in the following situation:

Problem:	Deliver package to work
Plan:	Drive car
Context:	May, midweek, morning (school time), pass in front of a school
Ext. Expect.:	Children in sight
Int. Expect.:	Reaching school zone
Action:	Drive straight, 25mph
$Time_S$:	max. 3 minutes

The experts were instructed to translate their qualitative feelings into quantitative values, and to concentrate more on relative values than on the absolute values they were giving. As some of them have commented, the scale used was sometimes closer to logarithmic and sometimes closer to exponential, but very seldom (if ever) was it approximately linear.

Each expert was also independently asked to order the set of contingencies by reaction value, that is, to specify the order in which he or she believes the agent should consider these contingencies for reaction, as well as where a threshold on monitoring for them should be placed. This information was then used to evaluate the results of applying our framework to the data supplied by the experts. The experts were asked to provide the contingency dependent knowledge independently of the final ordering. In any case, we believe there is little danger of any conscious correlation between the data supplied by an expert for each contingency and the order preference specified by the same expert, because of the amount of information they had to supply - over 50 values each only for contingency data.

I will omit here the individual values supplied by each expert for each contingency precisely because of the considerable amount of numbers

involved. I would prefer to comment instead a little on the distribution of these values, although a meaningful statistical analysis would not be fully relevant because the still small number of experts involved. The absolute values specified varied quite a lot. For example, the consequences of not reacting to the engine overheating was rated between 4 and 10 (on the scale of 0 to 10, where 0 meant no consequences at all), while the likelihood of a child running in front of the car was rated between 4 and 9. Although the ordering of the contingencies differed too (traffic light was placed between first and seventh while airplane, radio and meteor all varied between the ninth and the thirteenth places), the experts opinions agreed much more on the set of contingencies to be actually taken into account (i.e. the monitoring threshold): all of them indicated the first four contingencies as ordered in table 6.1, all but one indicated the hole, and all but two indicated the cat and the tire contingencies.

This was the first indication that, although individual pairs of experts may disagree, each of the experts tends to agree with the opinion of the group. This conjecture was then supported by a deeper analysis of the rest of the data supplied by the experts. We have further analyzed the order specified by the experts on the set of contingencies, by assigning an order number to each contingency according to each expert's specification, and then taking their median value, average value, and average of the set of 6 numbers obtained by eliminating the highest and lowest expert specified value for each contingency independently. In all three cases, the result of ordering the contingencies by the values obtained this way were identical, and the differences with each expert were much smaller than differences between individual experts. This again supports the previous conjecture. It was also interesting to see that not even one expert had specified the same ordering as inferred by all the three statistical methods. A further confirmation of the conjecture came from the fact that, for each characteristic of the contingencies, the three statistical measures have produced very similar results. Moreover, after eliminating the two extreme values in each case, the remaining values were much closer, which shows that the experts tend to agree with each other most of the time. Also, since different experts use different scales to measure the same qualitative phenomenon, the qualitative aspects of their input (orderings) tend to agree more than the quantitative

formulations (the experiments described further in this chapter will show that our framework is robust to quantitative variations in the knowledge specification, and is well suited to extract the qualitative aspects of it, which are the ones which eventually interest us).

The analysis of this data also suggested that different experts take the same (or consistent) decisions, but apparently for different reasons, that is, they have different heuristic "formulae" or rules to combine their evaluation of the characteristics of events in their domain of expertise. All these observations support our explicit inclusion in the framework of an expert model, which has the role of calibrating the entire reaction decision framework according to the set of qualitative_to_quantitative transformation functions used by the expert providing the domain knowledge.

Contingency		$Time_{rc}$	$Time_p$	Consequences	Side-effects	Likelihood
1	Child	1.0	10.0	10.0	6.5	4.5
2	Car-X	1.0	10.0	8.8	5.7	4.0
3	Car stop	2.0	5.0	7.0	3.2	6.2
4	Cat	1.0	10.0	5.2	5.9	6.8
5	T.light	4.0	2.5	6.2	0.7	8.8
6	Tire	3.0	3.3	6.0	2.8	2.3
7	Hole	2.0	5.0	4.5	4.5	2.8
8	Plane	2.5	4.0	9.5	4.5	0.3
9	Brake	30.0	0.3	6.2	1.0	2.0
10	Heat	50.0	0.2	5.5	0.3	2.2
11	Radio	100.0	0.1	1.8	0.7	2.0
12	Meteor	0.1	100.0	9.5	3.2	0.1
13	Ball	1.0	10.0	0.7	5.7	5.0

Table 6.2. Data values for the car driving domain experiments

In our experiments conducted with data from the driving domain, we have used the average_after_extremes_elimination values, obtained from the raw data provided by the experts as described above. These values are presented in table 6.2. The order in which the contingencies are presented in both tables 6.1 and 6.2 is the average_after_extremes_elimination (which, as mentioned above, is the same as the average and the median) order obtained from the pool of experts. The experiments with this set of data are briefly presented in the next two sections.

6.2. Optimality

We present here the results of the experiments we have conducted to support the theoretical claims made in chapter 5. Since most of these claims were justified theoretically, these experiments are merely demonstrations of applying the framework. We have used four different reactive planner models and five agent models to show how the recommendations of the framework vary and how it continues to ensure the optimal use of the agent's resources for the given agent models.

We have also used a "normal" behavior model, that is we expect the agent to behave the same way as the experts recommend. In calculating the reaction value of a contingency, this model assigns more weight to the time pressure dimension, followed by the difference between consequences and side-effects, and then likelihood. Consequences are taken into account both by themselves, but also (and mostly) in combination with the side-effects. Thus, the criticality function parameters given by the behavior model are:

$$p_1 = 5,\ p_2 = 1,\ p_3 = 0,\ p_4 = 0,\ p_5 = 3,\ p_6 = 2,$$

where the parameters specified by the expert model (an abstract, "average_after_extremes_elimination" expert) are:

$$T_{max} = 20.0;\quad T_{min} = 1.0;\quad CS_{min} = 2.3;\quad L_{min} = 1.3;\quad MON = 10000,$$

and the function computing the time pressure is:

$$f_{tc} = 10 / time_{rc}\ .$$

In this particular case, the criticality function (described in section 3.3.2) becomes:

Criticality = f_c (t, c, s, l) =

if	(t > 20)	*then*	$f_c = 0$
elseif	(c + 2.3 - s < 0)	*then*	$f_c = 0$
elseif	(t < 1)	*then*	$f_c = \sqrt{t^5 * c * (c+CS_{min}-s)^3 * l^2}$
elseif	(l < 1.3)	*then*	$f_c = \sqrt{t^5 * c * (c+CS_{min}-s)^3 * l^2}$
else			$f_c = t^5 * c * (c+CS_{min}-s)^3 * l^2$

Table 6.3 presents the values returned by this function, and the outcome of the monitoring decision of the framework. We provide them only to allow the reader to 'feel' the results of the framework. The monitoring threshold was set by the expert model in a region of the contingency space where there is a substantial gap among the reaction values of the contingencies ordered by criticality. Since the expert and behavior models do not change during the experiments described in this section, these values will not change either. They will however change anytime at least one of the expert or behavior models change. In the experiments described in the following sections we will not include this criticality value anymore. It can however be easily recomputed from the behavior models, which will always be specified.

Contingency		Criticality	Monitor
1	Child	3.95E9	yes
2	Car-X	2.21E9	yes
3	Car stop	1.90E8	yes
4	Cat	9.84E7	yes
5	Traffic light	2.22E7	yes
6	Tire	2.17E6	yes
7	Hole	1.34E6	yes
8	Plane	5.83E2	
9	Brake	6.56	
10	Heat	1.89	
11	Radio	5.3E-2	
12	Meteor	0.00	
13	Ball	0.00	

Table 6.3. Criticality values for the "normal" behavior model, for the car driving domain experiments

The first and most important observation of the experiment is that our framework orders the contingencies by criticality value (based on the data from the "average" expert) identically to the order indicated by the same "average" expert. When presented with this ordering, all the human experts involved have agreed to its rationality.

We must also point out here that the framework proved very robust, in that considerable variations in the values of the behavior and expert model parameters as well as in the absolute values for the dimensions of contingencies have yielded the same order induced by the criticality function. What really matters is the relative relationship among pairs of elements of the

framework. For example, in the normal behavior model, time pressure has greatest weight. We have experimented with variations of up to 25% in the absolute value of its weight (p_1) and have still obtained the same order. We have repeated the experiment for other behavior models were time pressure is also considered most important, as well as by varying other parameters or slightly varying individual values of the characteristics of contingencies, and in each case we have obtained robust behaviors of the framework. This suggests that small variations in the values provided by experts should not negatively influence the behavior of an agent using this framework.

In the experiments described in this section, we have used the following four reactive planner models:

RP1: constructs balanced binary decision trees; the function estimating the global reacting response time:

$$ft = k_p * \log_2 (\text{number\_of\_contingencies\_with\_}\geq\text{\_criticality}),$$

where the average test time is: $k_p = 0.2$ seconds.

RP2: same as RP1, but the average test time is: $k_p = 0.3$ seconds.

RP3: constructs decision lists; the function estimating the global reacting response time is linear:

$$ft = k_p * \text{number\_of\_contingencies\_with\_}\geq\text{\_criticality},$$

where the average test time is: $k_p = 0.2$ seconds, and the decision lists are built such that the pre-conditions discriminating the contingencies with the highest time pressure are tested first.

RP4: same as RP3, but the average test time is: $k_p = 0.3$ seconds.

We have also used five agent models. The only difference among them is the computational load estimated to be imposed on the agent at execution time (for this situation), which has the effect of slowing the agent, that is, it increases the response time of the agent to a contingency by a factor K_t:

$$f_{r_0}(\text{time}_r) = \text{time}_r * K_t\ ;$$

The five agent models used are:

AM1: $K_t = 1$, that is, there is no computational overhead estimated;

AM2: $K_t = 1.3$, that is, there is a 30% computational overhead estimated;

AM3: $K_t = 1.8$, that is, there is an 80% computational overhead estimated;

AM4: $K_t = 2.5$;

AM5: $K_t = 4.0$.

Contingency		Monitor	React (RPModel = decision trees - $k_p = 0.2$)				
			$K_t = 1.0$	$K_t = 1.3$	$K_t = 1.8$	$K_t = 2.5$	$K_t = 4.0$
1	Child	yes	yes	yes	yes	yes	yes
2	Car-X	yes	yes	yes	yes	yes	yes
3	Car stop	yes	yes	yes	yes	yes	
4	Cat	yes	yes	yes	yes	yes	
5	Traffic light	yes	yes	yes	yes		
6	Tire	yes	yes	yes	yes		
7	Hole	yes	yes	yes			
8	Plane						
9	Brake						
10	Heat						
11	Radio						
12	Meteor						
13	Ball						

Table 6.4. Optimality demonstrations results for reactive planner model RP1

Contingency		Monitor	React (RPModel = decision trees - $k_p = 0.3$)				
			$K_t = 1.0$	$K_t = 1.3$	$K_t = 1.8$	$K_t = 2.5$	$K_t = 4.0$
1	Child	yes	yes	yes	yes	yes	yes
2	Car-X	yes	yes	yes	yes	yes	
3	Car stop	yes	yes	yes	yes		
4	Cat	yes	yes	yes			
5	Traffic light	yes	yes	yes			
6	Tire	yes	yes				
7	Hole	yes	yes				
8	Plane						
9	Brake						
10	Heat						
11	Radio						
12	Meteor						
13	Ball						

Table 6.5. Optimality demonstrations results for reactive planner model RP2

Contingency		Monitor	React (RPModel = decision lists - k_p = 0.2)				
			$K_t = 1.0$	$K_t = 1.3$	$K_t = 1.8$	$K_t = 2.5$	$K_t = 4.0$
1	Child	yes	yes	yes	yes	yes	yes
2	Car-X	yes	yes	yes	yes	yes	
3	Car stop	yes	yes	yes	yes	yes	yes
4	Cat	yes	yes	yes			
5	Traffic light	yes	yes	yes	yes	yes	yes
6	Tire	yes	yes	yes	yes	yes	yes
7	Hole	yes	yes	yes	yes	yes	
8	Plane						
9	Brake						
10	Heat						
11	Radio						
12	Meteor						
13	Ball						

Table 6.6. Optimality demonstrations results for reactive planner model RP3

Contingency		Monitor	React (RPModel = decision lists - k_p = 0.3)				
			$K_t = 1.0$	$K_t = 1.3$	$K_t = 1.8$	$K_t = 2.5$	$K_t = 4.0$
1	Child	yes	yes	yes	yes	yes	
2	Car-X	yes	yes	yes			
3	Car stop	yes	yes	yes	yes	yes	yes
4	Cat	yes	yes				
5	Traffic light	yes	yes	yes	yes	yes	yes
6	Tire	yes	yes	yes	yes	yes	yes
7	Hole	yes	yes	yes	yes		
8	Plane						
9	Brake						
10	Heat						
11	Radio						
12	Meteor						
13	Ball						

Table 6.7. Optimality demonstrations results for reactive planner model RP4

Tables 6.4 to 6.7 summarize the results of our demonstrations. They list the set of contingencies recommended by our framework for reactive response preparation, in each case. As expected, this set decreases with an increase in the agent computational load, all other things being equal (different columns in the same table). It also decreases with an increase in the cost (here time) of the average tests to be performed (as can be seen by comparing the corresponding columns in tables 6.4 and 6.5, as well as the same

columns in tables 6.6 and 6.7. In each case, the agent tries to optimize the use of the agent resources (i.e. to include as many contingencies as possible), while maximizing the evaluation function on the subset of selected contingencies, by essentially including the highest criticality contingencies possible. Obviously, the more accurate the agent and planner models used, the better the selected contingencies will actually optimize the use of run-time resources (the models used here are quite rough - assume all tests take the same time and that the simple logarithmic and linear formulae stated above correctly approximate the agent).

In this example the decision trees model always selects the contingencies in the strict order of criticality (which need not be the case in general), while the decision lists model allows for gaps in the strict order, so that it can accommodate a larger number of contingencies. This is one more proof that the algorithm proposed in chapter 3 optimizes the use of the agent's resources. For example, in table 6.7, for $K_t = 2.5$, the agent can respond to only one contingency with a response time of maximum 1 second, so it chooses the one with largest criticality (the *child* contingency); it can respond to both contingencies with maximum response time of 2 seconds (the *car_stop* and the *hole* contingencies), and so on, but cannot respond to the other contingencies with short (1 second) response time, so it will omit them from the final set. Also note that the decision lists based planner model assumes that the contingencies are ordered by the response time allowed (in the final reactive plan), and also that the test times for each contingency are constant. If the first of these assumptions would have not been included in the reactive planner model, then the default assumption is that the contingencies are ordered by criticality, and then the reactive plan for this case could not have included the *hole* contingency since it would have been last in the decision list, and its response time would have exceeded its allowed response time.

One last observation from these experiments is that, for this particular set of data, it confirms our discussion of decision trees versus decision lists from section 3.4.1. We argued there that there are frequent cases in which the set of contingencies recommended by the framework is larger when using a decision lists planner than a decision trees planner, all other things being equal (which may seem somewhat counterintuitive at the first glance). Indeed, in this demonstration, the decision lists based agent includes more

contingencies than its decision trees based counterpart for most of the cases covered. In our example, the evaluation function value is usually greater in the decision trees case, because of a subtle violation of the "all other things being equal" assumption: the decision trees based planner model assumes that there is no test time needed to reach a response for a single contingency ($\log_2 1 = 0$), while the decision lists based planner model assumes that the time needed to reach such a response is still the time needed to perform one test. If this assumption would have been made in the first case too, then the decision lists planner model would have yielded also a higher evaluation function than its corresponding decision trees counterpart, for the set of contingencies recommended in at least some cases (like RP2 and RP4 ($k_p = 0.3$) and AM4 ($K_t = 4.0$)).

6.3. Behavior Models

Though not intended as a simulation of human behavior, our approach to solving the reaction planning decision problem has some potential applications in this area too. Specifically, it provides the basis for a possible language to discuss the characteristics of different human behavior models related to this task. In this section we shall propose a way of representing in our framework some such behavior models discussed in the literature, as well as the results of a few experiments we have done using this representation. Our discussion here is by no means intended to give a complete solution to the problem of simulating human reactive behaviors, but is only intended to suggest a possible such representation, which needs a lot more research to prove its usefulness or to find its best application domain.

In section 5.3, we have justified the property that our reaction decision framework consistently implements behavior models. We stated then the conjecture that for most types of reaction-related behaviors cited in the literature, there is a corresponding behavior model encoding in our framework which implements that type of reaction. Here, we go even a little further, by defining a couple more such behavior models and representing them in our framework too. Since we found no way to theoretically prove this conjecture, we have conducted a number of experiments designed to support it, which we present in this section. They show how our framework can

determine an agent to exhibit different reactive behaviors for the driving domain described before, while also helping us to clarify the meaning of the different thresholds and parameters in our framework.

Besides the so called "recommended" or "normal" behavior, we have found six more types of reaction-related behaviors - sometimes called hazardous attitudes [Woods & al., 1987; FAA, 1991]. The last two behaviors were proposed by David Gaba (personal communication, 1993). Here is a brief description of each of these behaviors:

- *Recommended Behavior* - is the normal behavior expected by the expert and from an expert in the domain.

- *Antiauthority Behavior* - is the "don't tell me!" type of behavior, in which the agent regards rules, regulations and procedures as unnecessary, and thus tends to disobey them.

- *Impulsivity Behavior* - is the "do anything quickly!" type of behavior, in which the agent attempts to always do the first thing that comes to mind, without stopping to think and select the best alternative.

- *Invulnerability Behavior* - is the "it won't happen to me" type of behavior, in which the agent is always inclined to take risks since it believes that the current situation is never one of those (less likely but still possible) situations when something wrong might just happen.

- *Macho Behavior* - is the "I can do it!" type of behavior, in which the agent wants to impress others, and is ready to take significant risks to do it. It is inclined to react even when not really necessary or when it may be more dangerous than not to react. Such agents either forget about the possible side-effects of their actions, or at least discount deeply these side-effects.

- *Resignation Behavior* - is the "what's the use?" type of behavior, in which the agent faced with a critical situation usually chooses to do nothing, since it underestimates its capacity to respond to the event and the effectiveness of such a response, in the given time frame. It has a tendency to leave such actions to others, for better or for worse.

- *Risk-averse Behavior* - the agent tries to avoid risk by all means (considering both the consequences of not being prepared to react in time, and the possible side-effects of reactions), but may therefore give sometimes less importance to the time pressure.

- *Liability Conscious Behavior* - the agent is particularly interested in avoiding any legal liabilities that may arise from its actions. Therefore, it tends to prepare to always do something, preferably what is legally bounding, even if that something may be believed not to succeed in that particular situation. This may prevent the agent to prepare for some other contingencies which are less liability creating, but which could have been treated if there were enough resources available.

- *Social Responsibility Behavior* - the "socially conscious" agent tends to put the interests of the society before those of the individual, including itself.

Each of these behaviors can be simulated in our framework by adjusting the parameters of the corresponding behavior model. While the actual parameter values are less important, their relative values define the different behavior models.

Behavior	Behavior Model						Expert Model			
	p_1	p_2	p_3	p_4	p_5	p_6	T_{max}	T_{min}	CS_{max}	L_{min}
Recommended	5	1	0	0	3	2	20.0	1.0	2.3	1.3
Antiauthority	5	1	0	0	3	0				
Impulsivity	0	0	0	0	0	3		10.0		5.0
Invulnerability	5	1	0	0	3	2				4.2
Macho	4	1	0	0	0	3			10.0	
Resignation	5	1	0	0	3	2	5.0			
Risk-averse	2	2	4	2	1	1				
Liability conscious	3	3	1	2	1	2	100.0	0.0		0.0
Social responsibility	4	3	0	0	4	3				

Table 6.8 Representing Behavior Models

Table 6.8 summarizes the representation of these behavior models into our framework. Recall that a behavior model in our framework is implemented by a set of values for the parameters of the criticality function (computing the reaction value of the contingencies), and may also be influenced by some

values of the thresholds given by the expert model (section 3.3). The values for the expert model parameters are completely specified in the table only for the recommended behavior model; for the other models, only values that have changed from the initial specification are given. Also remember that time pressure is the only parameter which can take values outside the interval [0,10], because it is converted from arbitrary real values using the expert specified conversion function f_{tc}. Therefore, the time pressure related parameters are harder to generalize among domains, as will be noticed in appendix 3, where we present the results of the same experiments run for the anesthesiology domain, with the same parameter values as here except for the time pressure dimension. The expert models in table 6.8 were used in our demonstrations in the driving domain.

To illustrate the simulation of these behaviors in our framework, we have run the framework with the behavior models presented in table 6.8 for the 13 contingencies presented previously the driving domain. Table 6.9 summarizes the results of these experiments. We have also shown the reaction values produced by the criticality function. Their absolute values have no meaning whatsoever; what matters are their relative values (and only within the same behavior model), which represent the relative value of reacting to one contingency vs. another in a same situation and under the same behavior model. For each behavior, the monitoring threshold was set (through the expert model) in a region of the contingency space where there is a substantial gap among the reaction values of the contingencies ordered by criticality. The threshold is represented by a thicker line separating the contingencies for each behavior into two sets. The numbering of contingencies for each behavior model is the same as for the recommended behavior. This was done in order to facilitate comparisons of each behavior model with the "normal" one.

In chapter 5, we have defined a behavior model to be an order relationship on the set of contingencies associated with a situation. Therefore, in the experiments described in this section, we only concentrate on the ordering of contingencies by reaction value (and sometimes relative values of the criticality function, but never on its absolute values), and ignore any issues related to the reactive planner model and the agent model, that is we ignore the final decision of applying the framework to a set of contingencies.

This is consistent with the purpose of our demonstrations here, since any specific agent (with a given reactive planner and resource limitations) may exhibit any of the reaction behaviors discussed, depending only on the order in which its behavior model recommends the contingencies for consideration to be reacted to, and not on the actual components and resources of the agent.

Behavior Model 1 (Recommended)			Behavior Model 2 (Antiauthority)			Behavior Model 3 (Impulsivity)		
1	Child	3.95E9	1	Child	1.95E8	4	Cat	3.14E2
2	Car-X	2.21E9	2	Car-X	1.38E8	5	T.light	26.10
3	Car stop	1.90E8	3	Car stop	4.96E6	3	Car stop	15.43
4	Cat	9.84E7	4	Cat	2.12E6	1	Child	9.54
6	T.light	2.22E7	6	Tire	4.10E5	2	Car-X	8.00
5	Tire	2.17E6	5	T.light	2.87E5	7	Hole	4.68
7	Hole	1.34E6	7	Hole	1.71E5	6	Tire	3.48
8	Plane	5.83E2	8	Plane	1.94E3	10	Heat	3.26
9	Brake	6.56	9	Brake	3.28	9	Brake	2.82
10	Heat	1.89	10	Heat	0.86	11	Radio	2.82
11	Radio	5.3E-2	11	Radio	2.6E-2	8	Plane	0.16
12	Meteor	0.00	12	Meteor	0.00	12	Meteor	0.00
13	Ball	0.00	13	Ball	0.00	13	Ball	0.00

Table 6.9 Reactive Behavior Experiments for the Driving Domain

Behavior Model 4 (Invulnerability)			Behavior Model 5 (Macho)			Behavior Model 6 (Resignation)		
1	Child	3.95E9	4	Cat	1.63E7	3	Car stop	1.90E8
3	Car stop	1.90E8	1	Child	9.11E6	5	T.light	2.22E7
4	Cat	9.84E7	2	Car-X	5.63E6	6	Tire	2.17E6
5	T.light	2.22E7	3	Car stop	1.04E6	7	Hole	1.34E6
2	Car-X	4.70E4	13	Ball	8.75E5	8	Plane	5.83E2
6	Tire	1.47E3	5	T.light	1.65E5	9	Brake	6.56
7	Hole	1.15E3	7	Hole	6.17E4	10	Heat	1.89
8	Plane	5.83E2	6	Tire	9.01E3	11	Radio	5.3E-2
9	Brake	6.56	8	Plane	8.10	1	Child	0.00
10	Heat	1.89	9	Brake	0.78	12	Meteor	0.00
11	Radio	5.3E-2	10	Heat	0.30	2	Car-X	0.00
12	Meteor	0.00	11	Radio	3.7E-2	4	Cat	0.00
13	Ball	0.00	12	Meteor	0.00	13	Ball	0.00

Table 6.9 Reactive Behavior Experiments for the Driving Domain (continued)

Behavior Model 7 (Risk-averse)			Behavior Model 8 (Liability conscious)			Behavior Model 9 (Social responsibility)		
1	Child	1.2E11	1	Child	2.0E11	1	Child	1.0E12
2	Car-X	3.7E10	2	Car-X	7.0E10	2	Car-X	3.7E11
4	Cat	4.39E9	12	Meteor	3.8E10	3	Car stop	7.0E10
3	Car stop	5.05E8	4	Cat	7.56E9	5	T.light	2.3E10
7	Hole	1.08E8	3	Car stop	3.34E9	4	Cat	2.89E9
6	Tire	2.40E7	5	T.light	7.49E7	6	Tire	2.96E8
5	T.light	1.88E5	7	Hole	7.48E7	7	Hole	3.49E7
8	Plane	1.59E5	6	Tire	5.04E7	8	Plane	4.10E3
9	Brake	57.62	8	Plane	3.17E7	9	Brake	2.72E2
10	Heat	2.33	9	Brake	1.37E4	10	Heat	94.70
11	Radio	0.57	10	Heat	4.87E2	11	Radio	0.78
12	Meteor	0.00	11	Radio	0.34	12	Meteor	0.00
13	Ball	0.00	13	Ball	0.00	13	Ball	0.00

Table 6.9 Reactive Behavior Experiments for the Driving Domain (continued)

Here is a brief explanation of the changes required by the parameters for each behavior model, with respect to the normal behavior model described in the previous section, as well as the main effects they have on the ordering of the 13 contingencies we have presented in the previous section, for the car driving domain:

- *Antiauthority Behavior Model* - do not take likelihood into account (most likely events are usually covered by laws, regulations and procedures). The *traffic light* contingency goes down in criticality, as the only regulation to be observed as a contingency in our set; the rest remains the same.

- *Impulsivity Behavior Model* - consider a single response, for a contingency with great (but serviceable) time pressure and high likelihood, to allow at least for a reasonable response in a significant number of cases; the reactive plan will consist of a single reaction to this contingency. Consequences and side-effects are disregarded, while time pressure is considered only through raising T_{min} (to 10) so as to include only the high but still acceptable time pressures. Likelihood is the only one still considered in the reaction value formula, and L_{min} is also raised significantly ($L_{min} = 5$). Therefore, the *cat* contingency becomes the only one selected for reaction preparation.

- *Invulnerability Behavior Model* - Low and medium likelihood contingencies are considered much less critical ("it won't happen to me..."); only high likelihood contingencies are really considered critical, so the L_{min} threshold is significantly increased (L_{min} = 4.2). In our tests, the *car crossing* contingency falls a lot because its likelihood becomes lower than this threshold.

- *Macho Behavior Model* - Forget about side-effects, and also take consequences less into account, since the agent mainly tries to impress others, by preparing for time pressured, but especially likely contingencies, so that it can react most of the time. The likelihood weight is increased, while the CS_{min} threshold is also increased (CS_{min} = 10.0) such that it becomes useless. In our demonstration, *ball* advances all the way to number 4 because the difference between consequences and side-effects is not considered here, while *cat* advances to number one since it is more likely than the first three, and its side-effects are also disregarded.

- *Resignation Behavior Model* - here it is interpreted as underconfidence, that is underestimating its own abilities, since we only talk about reaction preparation at planning time, and not reaction behavior at execution time (were it would have been interpreted as 'giving up'). The agent is willing to prepare to respond only to low time pressured events, and therefore the T_{max} threshold is significantly reduced (by 75% - T_{max} = 5). Therefore, many contingencies with higher time pressure get zero reaction value and fall at the end of the list.

- *Risk-averse Behavior Model* - taking most precautions to avoid risk, the decision process considers mostly the side-effects of the reaction, followed by the consequences of not reacting and the sum of consequences and side-effects, and much less time pressure and likelihood. The driving domain contingencies become roughly ordered by this sum, with a few exceptions: the *plane* contingency has very low likelihood, the *brake* contingency has very low time pressure, and the *meteor* contingency has a too short response time allowed for a reaction to be effective.

- *Liability Conscious Behavior Model* - while the weight of time pressure and of the difference between contingencies and side-effects decreases, the agent assigns more importance to consequences, side-effects and their sum. Also, there are no threshold for either time pressure or likelihood ($T_{min} = L_{min} = 0$; $T_{max} = 100$), since a contingency should never be discarded only because a reaction to it is believed to be useless. Therefore, *meteor* becomes very high priority here, and the agent will prepare roughly in the order of collision with people, moving cars, animals, objects. *Ball* is still not considered here at all because the side-effects are still much higher (and potentially more liable) than the consequences. Also in this case, more contingencies are considered for monitoring than usually.

- *Social Responsibility Behavior Model* - preparing a population optimal behavior involves considering both consequences alone and the difference between consequences and side-effects, as well as the likelihood, more than before (with respect to time pressure). It is closest to the "normal" behavior described in the previous section, with the only difference that *traffic light* gains priority with respect to *cat*, since this behavior tends to favor groups of people over single people, and people over animals. Notice here that significant overall changes in the values of the parameters, but small changes in their relative order, have produced a very similar ordering of the contingencies as compared to the recommended behavior.

As can be noticed from the above discussion, the results of these demonstrations require a certain amount of interpretation. This is necessary especially since the definitions of these behavior models are generally based on execution time types of reactions, while we attempt here to implement them at planning time. However, their interpretation shows that they are reasonable and consistent with the generally accepted (execution-time) definition of each behavior model, and that there is a plausible explanation for the results that maps them into the corresponding (conceptual) behaviors. These demonstrations show that our framework may at least provide a reasonable basis for representing and exchanging information and ideas about reaction-related behavior models, and thus for interpreting and studying different behaviors. For example, given a specific behavior (order on

the set of contingencies), we can automatically discover the parameters of a behavior model which emulates it, and then we can characterize this behavior and maybe attempt to correct it.

The specific values of the different parameters of the behavior models used may vary in certain limits, producing essentially the same results. This fact contributes to the robustness of our framework, and simplifies the knowledge acquisition process by easing the burden of specifying accurate values for the criticality space dimensions by the expert. More important are the relative values the expert supplies, but this is generally easier to acquire. Also the expert model may influence some of the behavior models, so the expert should probably be informed in advance about the desired behavior model. However, our experiments were conducted without informing the expert on the type of behavior model desired, and as can be seen from the discussion here (and also according to our experts), the results are in agreement with the definition of each behavior model.

We have also run the same demonstrations on a set of contingencies for a situation in the anesthesiology domain. Again the results satisfied the expert interpretation of the different behavior models. A brief description of this experiment and a short interpretation of the results for each behavior model are presented in appendix 3.

In the next section we present a final experiment, aimed at demonstrating that the framework defined in this thesis can scale up and be integrated in complex autonomous agents, designed to work in real, complex domains, and that by doing this, we improve the agent's global real time performance (by making it more responsive to those events that are considered more critical in the domain). This way we not only improve the quantitative performance of the agent, but more importantly, the quality of its performance. The experiment presented in the next section was also aimed as supporting evidence that the knowledge required to apply our framework exists in real domains, that it can be reasonably quantified by experts in the domain, and that it can be acquired from these experts and produce reasonable results.

6.4. Complex Real World Domain

We present here one more experiment we have conducted with our framework, in a real life medical domain: patient monitoring in an intensive care unit (ICU). This time, our framework was integrated in a complex real-time agent architecture capable of planning, reaction, and dynamic replanning: the Guardian system [Hayes-Roth & al., 1992, Hayes-Roth, 1990]. Our framework has the role of filtering the information which flows from the planner to the reactive planner, according to the architectural design outlined in appendix 1.

The two domain experts who have generously advised us (David Gaba and Serdar Uckun) have identified 68 contingencies for a set of situations corresponding to a general intensive care monitoring case (figure 6.1). They have also specified heuristic values for the four characteristics for each of these contingencies. For an easier understanding of the presentation, we shall present part of these experiments and most of the data concerned, in appendix 4, and shall discuss here only the main results.

Problem:	Intensive care monitoring
Plan:	normal postoperative procedure
Context:	after coronary artery bypass grafting (CABG) procedure, 50 years old patient, no other history known
Action:	ventilate patient / weaning / extubate patient
Internal Expect:	
External Expect:	
Time:	0-8 hours / 9-18 hours / 18-48 hours

Figure 6.1. Situations for the ICU domain

Table A4.1 lists the entire set of contingencies and the characteristic values for them, in the order specified by the experts (grouped by categories of complications that may develop).

The first part of this demonstration consisted in running the criticality function part of the framework on this data set, for the recommended behavior model (section 6.3), for several expert models. We have thus

exemplified for a large real-life case, the influence of varying different expert model parameters, over the ordering of contingencies by criticality. Appendix 4 presents a partial set of results from this demonstration (tables A4.2 to A4.5).

The most important conclusion to be drawn from this demonstration is that the recommendations of our framework are reasonable from the expert's point of view. Our experts have agreed, in each case (i.e. for each expert model used) with the ordering of the contingencies proposed by our system, finding them reasonable and finding reasonable interpretations for them. Since there is no other (objective) way to evaluate the framework's recommendations, we may conclude that the framework and the "normal" behavior model we have defined are a reasonable solution to our original problem.

#	Contingency (Response would be the typical response for this event)	Resp. time	Conse-quences	Side-eff.	Likeli-hood	Criti-cality
34	et-tube-disconnection	2	10	2	4	4.2E12
18	ventricular-tachycardia	1	9	7	2	2.2E12
13	ventricular-fibrillation	1	10	8	1	6.1E11
35	kinked-et-tube	5	8	2	4	1.8E10
20	hypoxia	5	8	6	4	2.53E9
7	myocardial-ischemia	5	8	6	3	1.42E9
15	sinus-bradycardia	5	7	5	3	1.24E9
14	ventricular-ectopy	5	7	7	6	7.62E8
5	cardiac-tamponade	5	8.5	7.5	3	6.84E8
19	sinus-tachycardia	10	6	5	7	8.21E7
22	cardiogenic-pulmonary-edema	10	8.5	7	3	3.26E7
1	myocardial-depression-post-cpb	10	8.5	7	3	3.26E7
32	pulmonary-embolism	10	8.5	7.5	3	2.13E7
6	hypovolemia	20	7	3	7	2.08E7
3	decreased-preload	20	7	3	7	2.08E7
25	pneumothorax	10	8	7	3	2.01E7
40	acute-hemolytic-transfusion-react	10	8.5	5	1	1.28E7
26	hemothorax	10	7	7	4	1.05E7
9	right-heart-failure	10	8	7	2	8.94E6
11	postop-hypertension	20	6.5	5	4	1.38E6

Table 6.10. Selected Contingencies for $k_p = 0.5$ (30 seconds) for ExplorerII ($k_t = 1.166$)

The second part of the demonstration considers the behavior of our framework in the context of the Guardian system. The blackboard-based

[Hayes-Roth, 1985] Guardian agent has a reactive planner (ReAct) using action-based hierarchies [Ash, 1994].

#	Contingency (Response would be the typical response for this event)	Resp. time	Conse-quences	Side-eff.	Likeli-hood	Criti-cality
34	et-tube-disconnection	2	10	2	4	4.2E12
18	ventricular-tachycardia	1	9	7	2	2.2E12
13	ventricular-fibrillation	1	10	8	1	6.1E11
35	kinked-et-tube	5	8	2	4	1.8E10
20	hypoxia	5	8	6	4	2.53E9
7	myocardial-ischemia	5	8	6	3	1.42E9
15	sinus-bradycardia	5	7	5	3	1.24E9
14	ventricular-ectopy	5	7	7	6	7.62E8
5	cardiac-tamponade	5	8.5	7.5	3	6.84E8
19	sinus-tachycardia	10	6	5	7	8.21E7
22	cardiogenic-pulmonary-edema	10	8.5	7	3	3.26E7
1	myocardial-depression-post-cpb	10	8.5	7	3	3.26E7
32	pulmonary-embolism	10	8.5	7.5	3	2.13E7
6	hypovolemia	20	7	3	7	2.08E7
3	decreased-preload	20	7	3	7	2.08E7
25	pneumothorax	10	8	7	3	2.01E7
40	acute-hemolytic-transfusion-react	10	8.5	5	1	1.28E7
26	hemothorax	10	7	7	4	1.05E7
9	right-heart-failure	10	8	7	2	8.94E6
11	postop-hypertension	20	6.5	5	4	1.38E6
4	increased-afterload	20	6.5	5	4	1.38E6
36	right-mainstem-intubation	20	6.5	3	2	1.23E6
16	atrial-fibrillation	20	7	6	4	9.78E5
41	febrile-nonhemolytic-transfus-react	20	6.5	4	2	6.98E5
67	low-k	30	7.5	5	5	6.63E5
42	mechanical-bleeding	20	7.5	7.5	4	3.54E5
66	dilutional-low-na	30	7	2	2	3.48E5
64	low-na	30	7	2	2	3.48E5
17	paroxysmal-supraventric-tachycardia	20	6	6	4	2.83E5
23	noncardiogenic-pulmonary-edema	20	8.5	8	2	1.81E5
68	high-k	30	8	7	4	1.47E5
31	bronchospasm	30	8	7	4	1.47E5
62	low-mg	60	7	3	7	8.57E4
45	intrinsic-pathway-defects	60	7	3	5	4.37E4

Table 6.11. Selected Contingencies for $k_p = 0.5$ (30 seconds) for SPARC10 ($k_t = 1.02$)

The reactive planner model for it (kindly specified by my colleague and its designer, David Ash) states that the reactive plan built tends to be an implicit hierarchy with about 3 levels, with a roughly constant branching factor throughout. Actually distinguishing a child node in the implicit

hierarchy is accomplished in the real hierarchy with a decision list-like structure. According to this model, reaching a contingency in the plan built for n contingencies takes roughly a constant time, equal to $3 * \sqrt[3]{n}$ times the amount of time for a single test (assuming the tests take approximately constant time). This assumption can be made in our domain and for our agent, since tests which take much longer (e.g. laboratory tests) are to be included in the main plan by the planner, to be performed regularly so that their data is always meaningful. This is generally the way physicians operate in real ICU settings. Therefore, for the purpose of our model, we can assume that the length of a test is roughly given by the time a human operator needs in order to retrieve and check a piece of data and to input it into the computer, i.e. approximately 30 seconds. The reactive planner model also allows for a small set of contingencies (say, three) to be hooked directly to the top of the hierarchy, and thus to be reached by tests independently of the other contingencies to be solved by this reactive plan. This is useful when there are a few very time critical contingencies, and the rest are with a much smaller time pressure.

#	Contingency (Response would be the typical response for this event)	Resp. time	Conse-quences	Side-eff.	Likeli-hood	Criti-cality
34	et-tube-disconnection	2	10	2	4	4.2E12
18	ventricular-tachycardia	1	9	7	2	2.2E12
13	ventricular-fibrillation	1	10	8	1	6.1E11
35	kinked-et-tube	5	8	2	4	1.8E10
20	hypoxia	5	8	6	4	2.53E9
7	myocardial-ischemia	5	8	6	3	1.42E9
15	sinus-bradycardia	5	7	5	3	1.24E9
14	ventricular-ectopy	5	7	7	6	7.62E8
5	cardiac-tamponade	5	8.5	7.5	3	6.84E8
19	sinus-tachycardia	10	6	5	7	8.21E7
22	cardiogenic-pulmonary-edema	10	8.5	7	3	3.26E7

Table 6.12. Selected Contingencies for $k_p = 0.6$ (36 seconds) for ExplorerII ($k_t = 1.166$)

The agent model only takes into account the slowdown of the system due to computational overhead. Simulations on two different platforms have yielded significantly different results: if Guardian is run on ExplorerII machines, the computational overhead is on average 16% for the simulated time period we are interested in (approximately two hours of simulated time);

on a SPARC10 workstation, this overhead is reduced to approximately 2%. Table 6.10 presents the results of running our entire framework, with the reactive planner and agent models described here, for the Guardian agent running on an ExplorerII platform. Table 6.11 presents the same results for a SPARC10 workstation. We have run the same experiment for an estimated test time of 20% larger (36 seconds) and the results are presented in tables 6.12 and 6.13 for ExplorerII and SPARC10 respectively. In the second case, the system was able to include about 75% more contingencies in the reactive plan. Also note that in all cases the system was able to include about 66% more contingencies in the reactive plan to be run on the SPARC10.

#	Contingency (Response would be the typical response for this event)	Resp. time	Conse-quences	Side-eff.	Likeli-hood	Criti-cality
34	et-tube-disconnection	2	10	2	4	4.2E12
18	ventricular-tachycardia	1	9	7	2	2.2E12
13	ventricular-fibrillation	1	10	8	1	6.1E11
35	kinked-et-tube	5	8	2	4	1.8E10
20	hypoxia	5	8	6	4	2.53E9
7	myocardial-ischemia	5	8	6	3	1.42E9
15	sinus-bradycardia	5	7	5	3	1.24E9
14	ventricular-ectopy	5	7	7	6	7.62E8
5	cardiac-tamponade	5	8.5	7.5	3	6.84E8
19	sinus-tachycardia	10	6	5	7	8.21E7
22	cardiogenic-pulmonary-edema	10	8.5	7	3	3.26E7
1	myocardial-depression-post-cpb	10	8.5	7	3	3.26E7
32	pulmonary-embolism	10	8.5	7.5	3	2.13E7
6	hypovolemia	20	7	3	7	2.08E7
3	decreased-preload	20	7	3	7	2.08E7
25	pneumothorax	10	8	7	3	2.01E7
40	acute-hemolytic-transfusion-react	10	8.5	5	1	1.28E7
26	hemothorax	10	7	7	4	1.05E7
9	right-heart-failure	10	8	7	2	8.94E6

Table 6.13. Selected Contingencies for $k_p = 0.6$ (36 seconds) for SPARC10 ($k_t = 1.02$)

The sets of selected contingencies include the first as many as possible contingencies in the order of their criticality value (table A4.2). They do not include the fourth contingency in table A4.2 because of the special treatment of highly time pressured contingencies in the reactive planner model (it allows for three contingencies to be reacted to faster than the rest - otherwise, the set of contingencies might have included only the first four

contingencies, but very few others if any). Due to the decision tree form of the reactive plan, all leaves are reached in approximately the same time (section 3.4.1), so the set of contingencies selected is limited by the response time to the most time pressured contingency included (in our case 5 minutes, since the one and two minute contingencies are treated separately).

These experiments reinforce a few statements we have made along the thesis. They show that the framework proposed here is useful in pruning the set of contingencies for which the agent should prepare to react. This is however necessary only in such domains where the number of contingencies is large enough to pose problems due to agent resource limitations (and we have characterized such domains in chapter 2); Guardian and its domain are typical in this respect. The performance of the enhanced agent improves upon the performance of the same agent without the benefit of our framework, because in the latter case, the reactive planner would have prepared a reactive plan to include all 68 contingencies, and due to its size, the resource requirements for such a plan could not achieve reactions to the most time pressured contingencies in this set. The set of contingencies selected depends on the characteristics of the agent and of its reactive planner (as represented by the agent model and reactive planner model). The more accurate these models are, the better will be the use of agent resources made by the set of contingencies selected. Also note that the agent may exhibit different reactive behaviors, as defined by the reactive behavior model.

Our experiments also show that the necessary data for our framework to be applicable exists in practice and can be acquired from experts in real-world domains. The more difficult part of the knowledge acquisition process was the identification of the set of contingencies possible in a given situation (the acquisition of the characteristic values for them was much easier, especially since their absolute values are less important than their relative order, due to the robustness of the framework).

The experiments described in this chapter and performed in different domains requiring quite different types of human expertise (mundane tasks, highly skilled domains, etc.) demonstrate the applicability of our framework in the general types of domains described in chapter 2.

Chapter 7
Conclusions

Most research projects have their roots in one or two basic questions, attempt (more or less successfully) to provide answers to these questions, and during this process usually generate many more new questions than answers. This thesis was no exception. In the next section, we present a summary of the answers which our work provides, and in the following section we enumerate a few questions raised and research avenues opened during our efforts to find solutions to the original problems stated in chapter 2.

7.1. Summary

Executing plans in the real world has long ago been recognized as a difficult and uncertainty-filled problem, due to contingencies generated by interactions between the executing agent and its environment. Conditional planning, reaction and dynamic replanning are all possible control modes to solve this problem, but none of them alone is entirely suitable for agents with limited resources working in complex environments. Therefore, the need arises for a mechanism to select, from the set of possible contingencies in the domain, the subsets which should be treated using each of the previously mentioned control modes. In this thesis we have defined a framework to select the subset of contingencies which are best suited for reactive response. Our framework's decisions are based on the plan situation under consideration, the characteristics of the contingencies and of an expert model specifying them,

as well as on the reactive planner and agent models. A behavior model determines the type of reactive behavior to be exhibited by the agent. All these models are designed as application-dependent plug-in modules to be attached to our framework, thus substantially increasing its generality and applicability across domains and types of agents. The decision of whether to prepare a reaction to a given contingency or not is taken while considering the entire set of contingencies that may appear in that situation, in relationship with the limitations of the agent's execution time resources. We have justified a few theoretical claims about our framework (including the optimal use of agent's resources), and then we have verified them experimentally. We have also demonstrated other properties of the framework, the most important being that the reactive behavior of an agent using our framework has the agreement of the experts in the field.

A couple of extensions to our framework were also discussed. The first one involves a similar framework to decide on the subset of contingencies for which to prepare a conditional branch (all the way to the final goal) in the plan. The second involves a proposal for a knowledge representation formalism for the types of knowledge involved in our framework: contingencies, reactions and situations. It was designed to facilitate the structuring and manipulation of this knowledge, as well as to facilitate the use of automatic knowledge acquisition and learning techniques to cope with the explosion of the related knowledge in complex domains. However, both these extensions were discussed only at a theoretical level and, as stated in the next section, they need more work in order to be fully understood and for their potential to be fully used.

7.2. Future Work

It is unfortunate (or maybe actually very fortunate) that a thesis cannot encompass an entire research career. Unfortunate because while trying to solve the originally stated problems, there are so many new problems that arise and which I would have liked to address. Fortunate because I am sure that while trying to address these new issues, many other problems would arise, and then no thesis would ever be finished. We shall briefly overview in this

section a few of the research issues which came up while solving the problems mentioned before.

Two already stated issues are the extensions to our framework cited above. The first involves the framework for deciding whether to prepare a full conditional branch for a contingency. While we have defined the general framework in section 3.5, there are many details that still have to be sorted out before a usable framework like the one for reactions can be obtained. The function computing the conditional planning value of a contingency must be identified and tested, and the values for its parameters must be specified for a normal behavior model (and possibly for other types of behavior models). Guidelines for specifying the planner model and especially the agent model (from the perspective of conventional plan execution) must also be set.

The second issue involves the knowledge representation formalism proposed in chapter 4. Since specifying the nonterminals of the grammar imposes some additional burden on the experts, it would be very helpful to devise a set of knowledge acquisition and learning tools to help the expert in this task. We believe that the best results here can be achieved by combining automatic learning methods with interactive knowledge acquisition tools (similarly with the methods used in [Dabija & al., 1992a]). Such an approach would better use the potential for bias shifting [Utgoff, 1988] and concept classification that this knowledge representation formalism is appropriate for.

Another open research issue related to our framework is its potential integration with case based reasoning and planning techniques. Figure 7.1 presents the possible information flow in such a system. The agent's knowledge base (contingencies and associated reactions in specific situations) may be organized as a library of cases. The agent may also have a library of reactive plans already built (each reactive plan built, may be cached into this library), organized by the situations in which they may apply. New knowledge may be added at any time to the case library, and each time an already encountered situation arises, the reactive plan that may already exist in the plan library is combined with any new contingency-reaction pairs applicable in that situation that have been included in the agent's knowledge base since the last use of this reactive plan. Our framework will decide, for each such situation, which are the best contingencies for which reactions should be

included in the updated reactive plan. If no new relevant knowledge (i.e. applicable in the current situation) has been added to the knowledge base since its last use, this reactive plan may be used without any changes or delay. Many issues arise here related to the independent management of the two libraries (knowledge structuring, and "forgetting") as well as the relationships between them. There are also interesting research issues related to the problem of acquiring the knowledge for the two libraries: knowledge for each of them may be acquired from an expert (and here interactive knowledge acquisition techniques may be used) or from the agent's own domain experience.

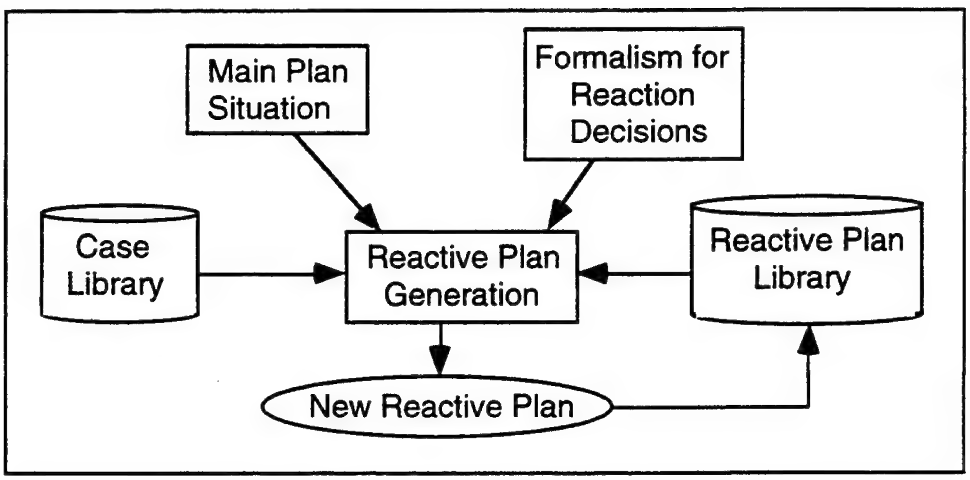


Figure 7.1. Extended system architecture

In domains where strong theories about possible contingencies exist, these theories can be used to anticipate all the contingencies that may appear for situations along the plan, and to specify their characteristics. However, in most domains with which we are concerned, such theories either do not exist, or they are very weak (e.g. cover the domain only partially, or can anticipate only certain kinds of events all over the domain). In such cases, the agent may generate prototype cases (akin to the cases in the case library) and propose solutions for them. They may then be evaluated and compared to corresponding actual cases, and the differences may be used to improve the weak domain theory that has generated them in the first place.

In this thesis we have also introduced a formalism to describe reactive behavior models. As we have shown in chapter 6, most of the human reactive

behavior models described in the literature can be conveniently expressed in our framework, which therefore provides a possible vehicle for the exchange of information on this subject. However, we have only touched the tip of the iceberg in this respect. Considerably more research is needed to refine this formalism so that it can be really useful for providing complete characterizations of these behavior models and therefore become useful in attempts to correct or influence human behaviors in critical domains like nuclear power plant supervision or aircraft flying. For example, in order to better model the differences between behaviors like *social responsibility* and *individualism*, the consequences dimension of the criticality space may be split into two components: (i) internal-consequences (which directly affect our agent) and (ii) external-consequences (effects of not responding to the contingency, over other agents in the environment).

As stated at the beginning of this section, the range of open problems suggested by this research is very wide, and we believe that at least part of them are worth further investigation.

Appendix 1
System Architecture

We briefly present here the way our framework is to be integrated in the general architecture of an agent with planning, reaction and monitoring capabilities. We assumed a modular system, in which each component can, in principle, be plugged in and out and the agent's performance should change gracefully. For example, if the agent is to operate without a reactive planner, then it will be able to respond only to the contingencies for which conditional branches have been prepared by the planner, while if it is to operate only with a reactive planner, then the agent should be able to react to all the contingencies for which it has reactions prepared for, but may never reach the overall goal since it lacks the main plan to do it. The framework to decide whether to prepare to react may be regarded as another such module, which when present, ensures that the agent is better prepared to cope with the different contingencies that may appear during its plan execution.

An alternative view is that the other agent modules (the planner, reactive planner, execution mechanisms, knowledge base, the expert model and the behavior model) are all independent modules which can be plugged into, and out of, the framework discussed in the thesis. The framework was defined in a general manner such that all these modules are parameters which will change the outcome of the analysis, but the general principles presented in chapters 3 and 4 and the theoretical analysis in chapter 5 remain all valid (since they all were done independent of any particular such module).

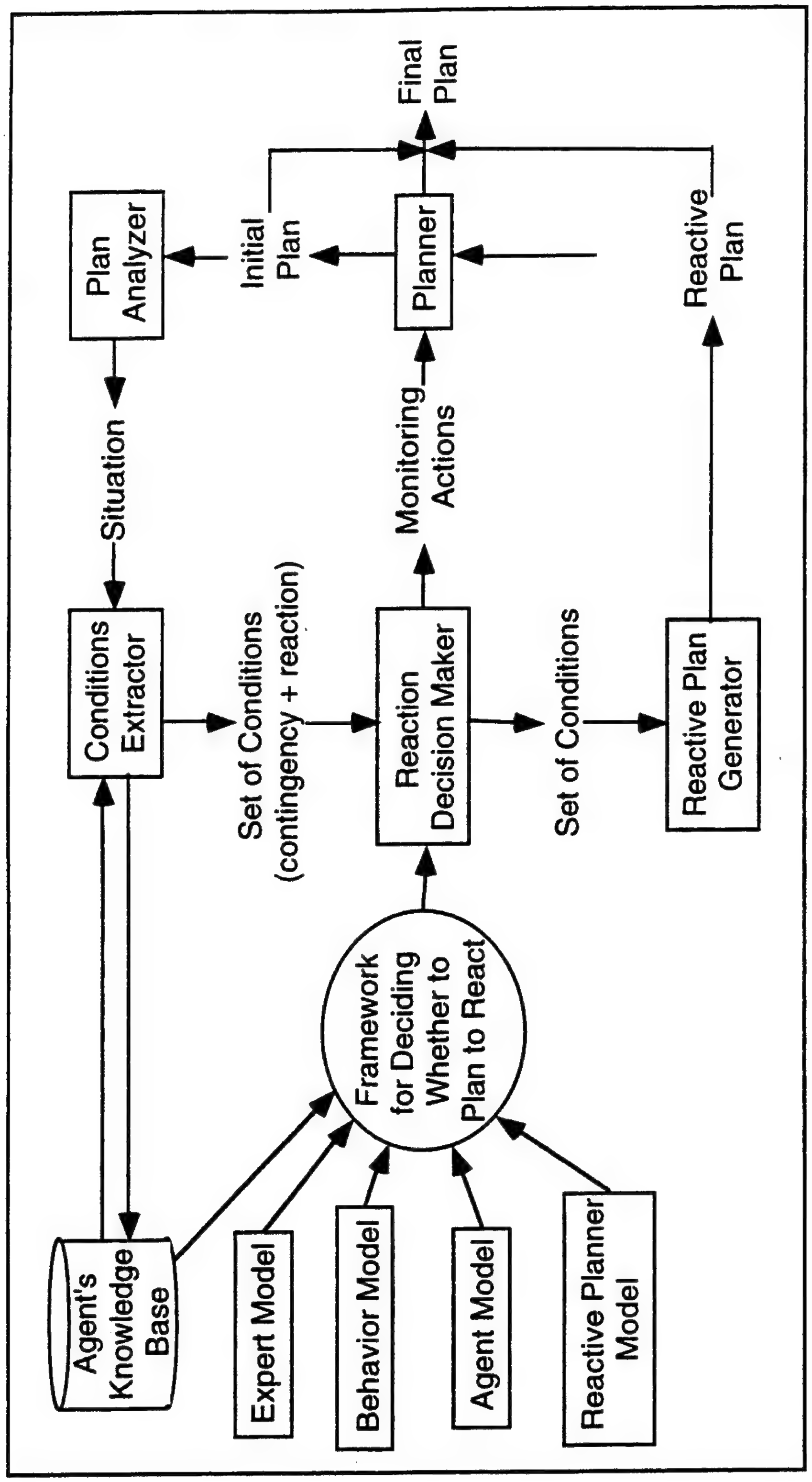


Figure A1.1. System Architecture and Information Flow

Figure A1 presents how all these modules fit together in a "complete" agent, as well as the information flow during the plan modification process. We assume this process starts when the *planner* has produced a complete (conditional) plan to solve a given problem. In order to identify the situations that may generate contingencies in the plan, the *plan analyzer* scans the plan and for each stage (situation) searches the *agent's knowledge base* for the set of contingencies that may appear in that situation, and their appropriate reactions. Each situation for which there are known contingencies will be further analyzed to prepare reactive plans for it.

All relevant contingencies found in the *agent's knowledge base* by the *contingency extractor* for a certain situation are passed on to the *reaction decision maker* which uses our framework presented in chapter 3, together with an *expert model*, a *behavior model*, the *agent model* (corresponding to the execution capabilities of this agent), and the *reactive planner model* (corresponding to the reactive planner available to this agent), to select those contingencies for which reactive responses should be prepared by the *reactive plan generator*. The reactive plan is passed back to the *planner* together with monitoring actions to be included in the plan. The reactive plan is eventually attached to the context-specific plan and the next stage of the plan will be subsequently analyzed.

This entire process is performed first at planning time, before the agent starts executing the main plan, and is repeated each time the agent is forced to dynamically replan its actions (and generate a new main plan) during the execution phase because of a major failure in executing the initial main plan.

One agent with such an architecture with which we have conducted demonstrations of our framework is the Guardian agent (for monitoring patients in an intensive care unit) [Hayes-Roth, 1990]. The results of these demonstrations are discussed in section 6.4.

Appendix 2

Knowledge Representation in the Car-Driving Domain

We continue here the example started in section 4.2, with the hierarchical vocabularies and the corresponding grammars for representing reactions and situations in the car driving domain. While we do not plan to specify the complete vocabularies for this domain, the ones that are given here are sufficient to represent all the examples encountered in chapter 3, as well as the experiments discussed in chapter 6 for the driving domain. They are also enough to represent a good deal more knowledge from this domain.

Figure A2.1 presents the hierarchical vocabulary for representing reactions in the car driving domain. This hierarchy is equivalent (according to the formalism discussed in chapter 4) to the following grammar:

G = (N, T, P, S), where:

N = { Reaction, Brake, Steer, Other, Left, Right, Hard, Gently,
Adjust_Radio }

T = { B.Hard, B.Gently, B.None, Left&Hard, Right&Hard, Left&Gently,
Right&Gently, None, Turn_on_Lights, Adjust_Volume,
Adjust_Station, Open_Window }

P = { Reaction -> Brake - Steer | Other
Brake -> B.Hard | B.Gently | B.None

Steer -> Left | Right | Hard | Gently | None
Left -> Left&Hard | Right&Hard
Right -> Right&Hard | Right&Gently
Hard -> Left&Hard | Right&Hard
Gently -> Left&Gently | Right&Gently
Other -> Turn_on_Lights | Adjust_Radio | Open_Window | ...
Adjust_Radio -> Adjust_Volume | Adjust_Station }

S = Reaction

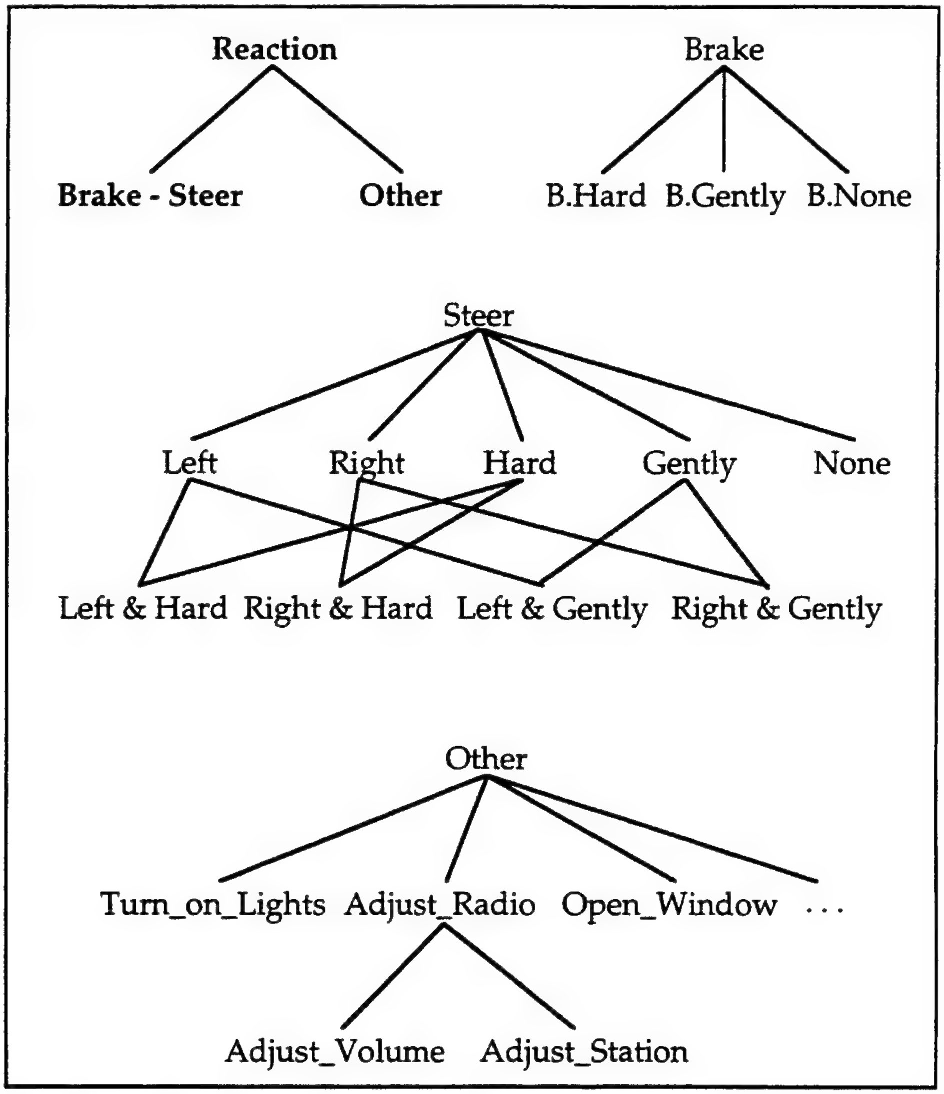


Figure A2.1. Vocabulary for describing reactions in the driving domain

Every reaction specified in table 3.1 can be obtained through a number of different derivations in this very small and simple grammar. Also, many

other reactions in the driving domain can be expressed using this vocabulary (this is generally true especially for reactions, since there are usually a small set of actions in a domain which can make up useful reactive plans in that domain). Since general reactions are often enough to be specified, the derivations may be stopped at those levels where the reaction expressed by the sentential form obtained thus far "contains" (according to the order relation defined in chapter 4) all the elementary reactions acceptable in that situation. For example, if the agent only needs to reduce speed somewhat, than "Brake" may be sufficient, without qualifying the action further.

Here is an example of deriving the reaction "Brake hard and steer right" to the first contingency in table 3.1 ("Child runs from right 20m in front of car"):

Reaction ->
Brake - Steer ->
B.Hard - Steer ->
B.Hard - Right.

This derivation has already been stopped before reaching a sentential form made up only of terminals in the vocabulary, since the "Right" nonterminal could have been further refined to one of the two terminals given by the production:

Right -> Right&Hard | Right&Gently.

It therefore represents a set of possible reactions, contained in this description (i.e. derivable from it).

Figure A2.2 presents the hierarchical vocabulary for representing situations in the car driving domain.

Some productions (both shown in figure A2.2 and omitted) may be realized through identification functions, as shown in chapter 4. For example, the grammar symbols Slow, Medium, Fast, can be considered nonterminals (instead of terminals like in this example), and the actual values of the speed can be considered terminals.

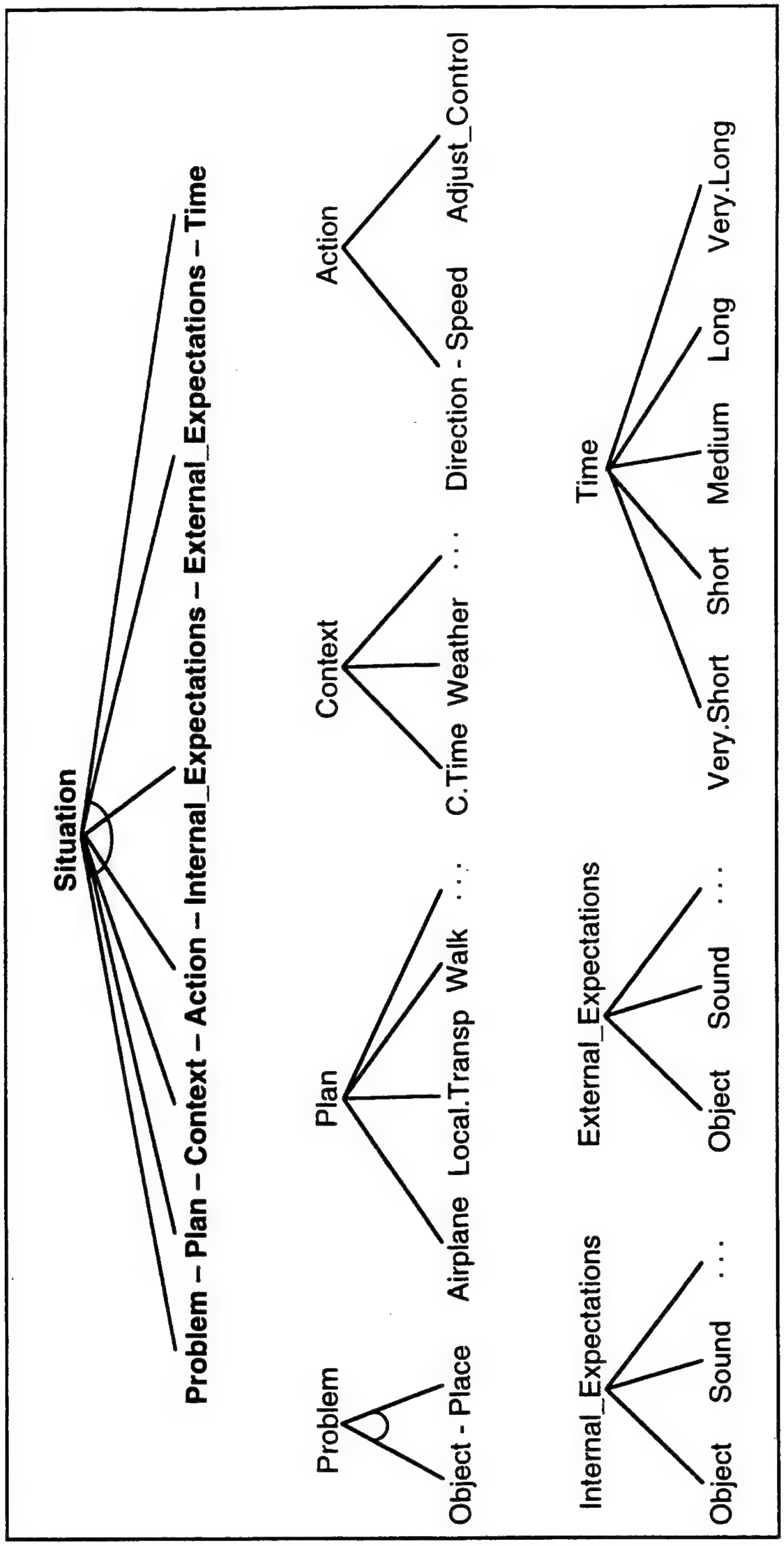


Figure A2.2a. Vocabulay for describing situations in the driving domain

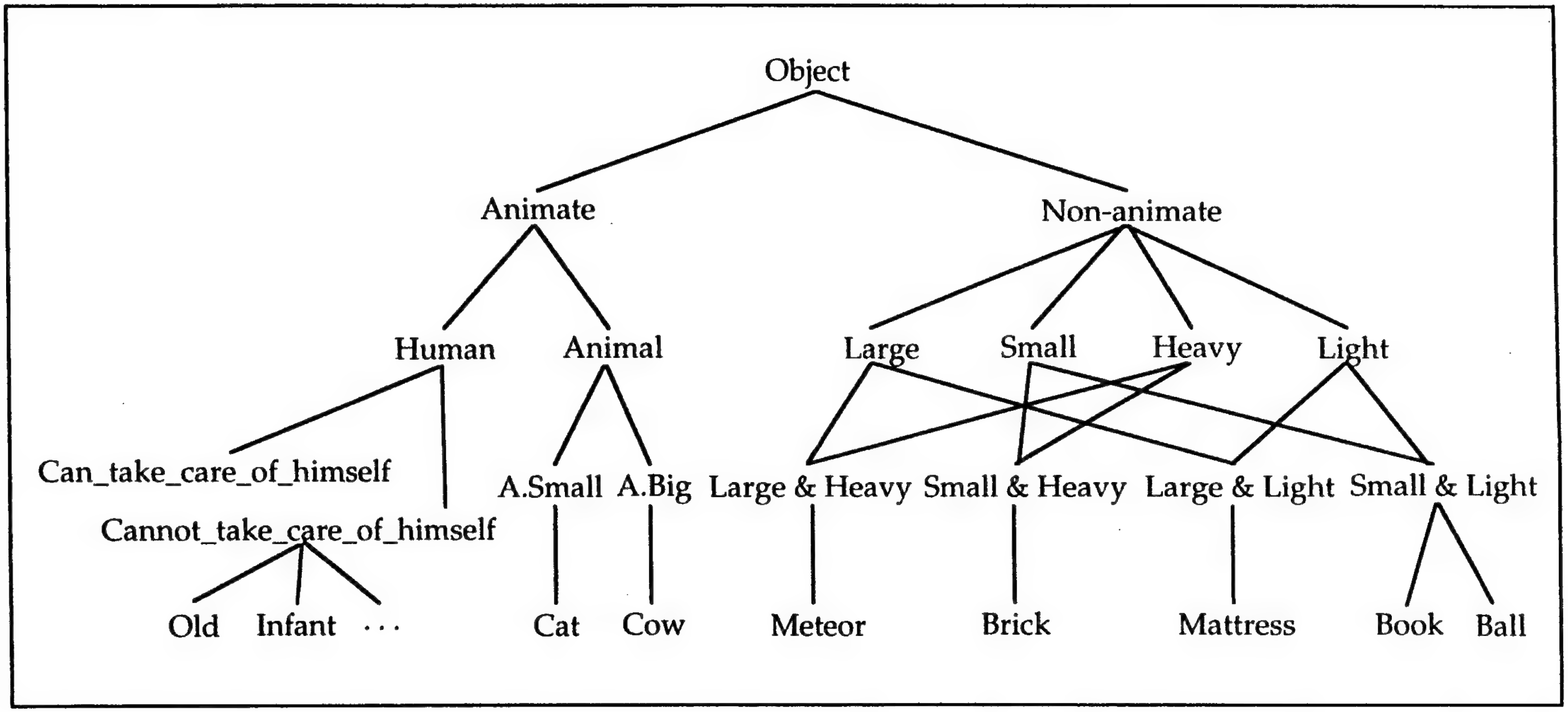


Figure A2.2b. Vocabulay for describing situations in the driving domain (continued)

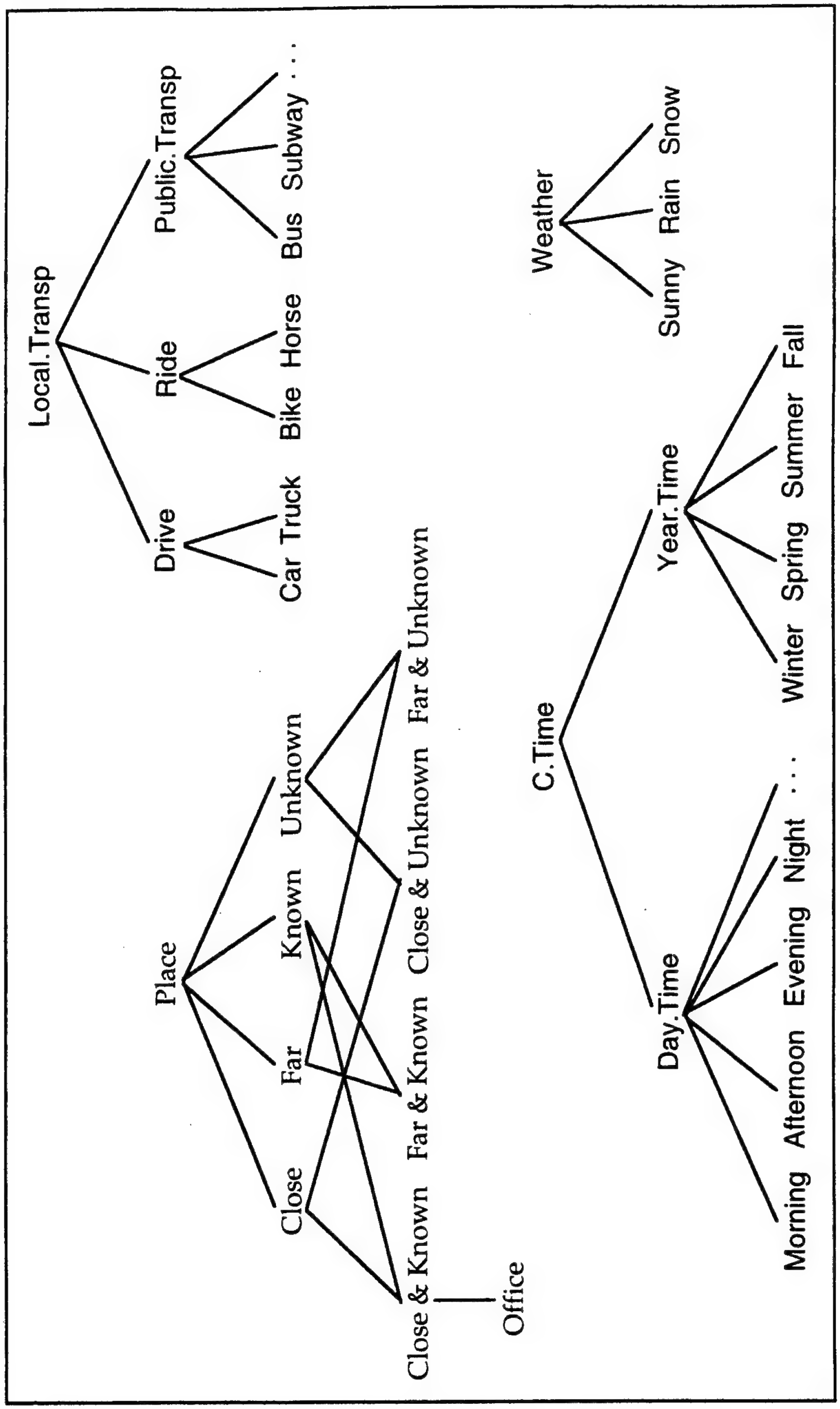


Figure A2.2c. Vocabulay for describing situations in the driving domain (continued)

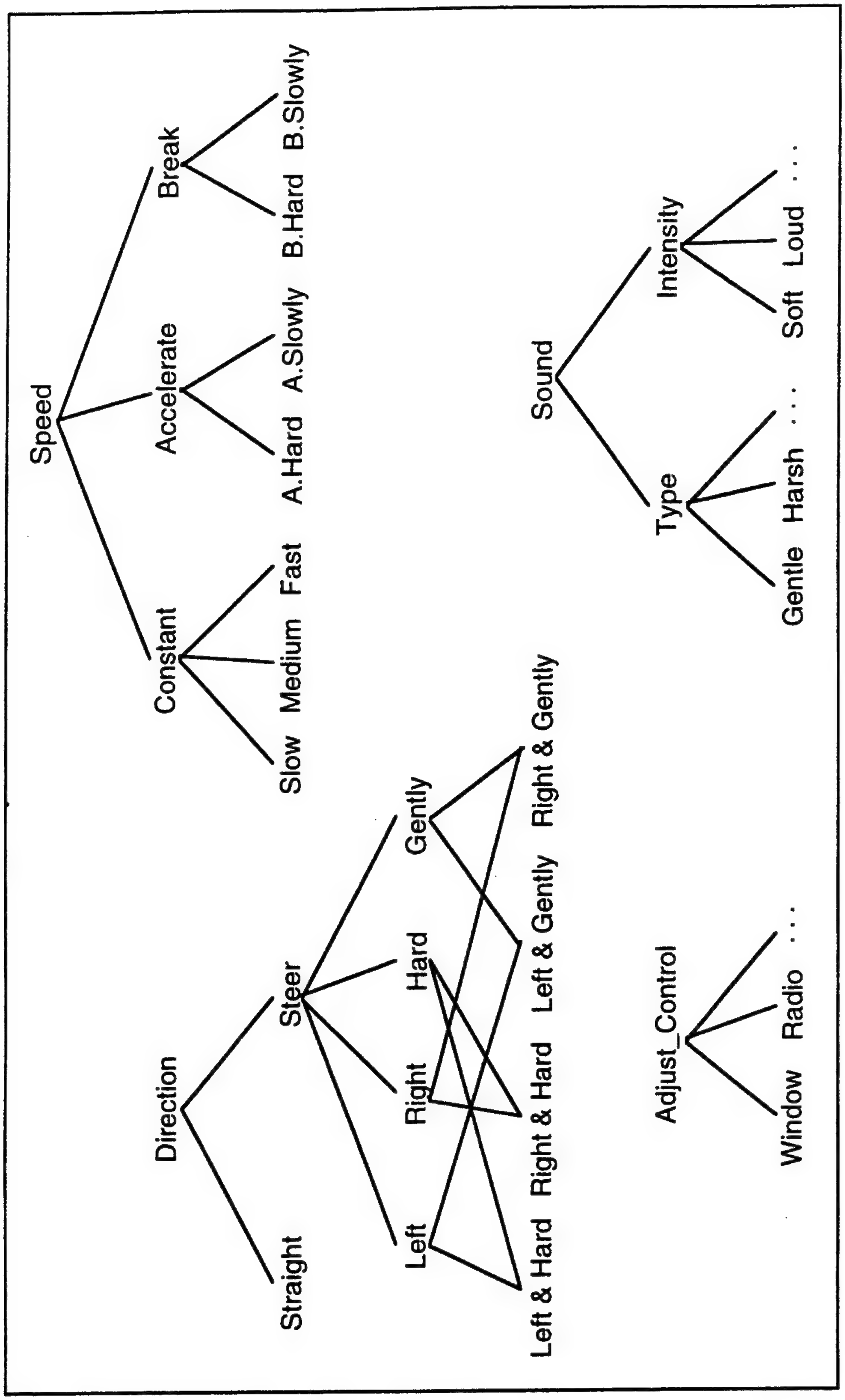


Figure A2.2d. Vocabulay for describing situations in the driving domain (continued)

An example of such a function may then be:

$$\text{Slow} = f_S\ (\text{speed}) = 5\ \text{mph} < \text{speed} < 20\ \text{mph},$$

which can be used to perform the transition over the edge linking "Slow" with the actual terminal, say "speed = 15 mph".

We have collapsed the seven vocabularies for representing values for the seven dimensions of the situation space into a single vocabulary, with the help of the first production of the grammar. Alternatively, we could have specified seven independent grammars, by throwing out the first production and the nonterminal Situation; each of these grammars would have had as starting symbols the nonterminals: Situation, Problem, Plan, Context, Action, Internal_Expectations, External_Expectations, Time (respectively), as productions all the productions which can be reached from their respective start symbols using the productions of the reunited grammar, and as nonterminals and terminals all those from the large grammar which are involved in the productions of each respective grammar.

The hierarchy in figure A2.2 is equivalent (according to the formalism discussed in chapter 4) to the following grammar:

G = (N, T, P, S), where:

N = { Situation, Problem, Plan, Context, Action, Internal_Expectations,
External_Expectations, Time,
Object, Animate, Human, Cannot_take_care_of_himself, Animal,
A.Small, A.Big, Non-animate, Large, Small, Heavy, Light,
Large&Heavy, Small&Heavy, Large&Light, Small&Light,
Place, Close, Far, Known, Unknown, Close&Unknown,
Local.Transp, Drive, Ride, Public.Transp,
C.Time, Day.Time, Year.Time, Weather,
Direction, Steer, Left, Right, Hard, Gently,
Speed, Constant, Accelerate, Break,
Adjust_Control,
Sound, Type, Intensity }

T = { Airplane, Walk,

Very.Short, Short, Medium, Long, Very.Long,
Can_take_care_of_himself, Old, Infant, Cat, Cow, Meteor, Brick,
Mattress, Book, Ball,
Office, Far&Known, Close&Unknown, Far&Unknown,
Car, Truck, Bike, Horse, Bus, Subway,
Morning, Afternoon, Evening, Night,
Winter, Spring, Summer, Fall,
Sunny, Rain, Snow,
Straight, Left&Hard, Right&Hard, Left&Gently, Right&Gently,
Slow, Medium, Fast, A.Hard, A.Slowly, B.Hard, B.Slowly
Window, Radio,
Gentle, Harsh, Soft, Loud,
. . . }

P = { Situation -> Problem - Plan - Context - Action -
Internal_Expectations - External_Expectations - Time
Problem -> Object - Place
Plan -> Airplane | Local.Transp | Walk | . . .
Context -> C.Time | Weather | . . .
Action -> Direction - Speed | Adjust_Control
Internal_Expectations -> Object | Sound | . . .
External_Expectations -> Object | Sound | . . .
Time -> Very.Short | Short | Medium | Long | Very.Long
Object -> Animate | Non-animate
Animate -> Human | Animal
Human -> Can_take_care_of_himself |
Cannot_take_care_of_himself
Cannot_take_care_of_himself -> Old | Infant | . . .
Animal -> A.Small | A.Big
A.Small -> Cat | . . .
A.Big -> Cow | . . .
Non-animate -> Large | Small | Heavy | Light
Large -> Large&Heavy | Large&Light
Small -> Small&Heavy | Small&Light
Heavy -> Large&Heavy | Small&Heavy
Light -> Large&Light | Small&Light

Large&Heavy -> Meteor | . . .
Small&Heavy -> Brick | . . .
Large&Light -> Mattress | . . .
Small&Light -> Book | Ball | . . .
Place -> Close | Far | Known | Unknown
Close -> Close&Known | Close&Unknown
Far -> Far&Known | Far&Unknown
Known -> Close&Known | Far&Known
Unknown -> Close&Unknown | Far&Unknown
Close&Unknown -> Office | . . .
Local.Transp -> Drive | Ride | Public.Transp
Drive -> Car | Truck
Ride -> Bike | Horse
Public.Transp -> Bus | Subway | . . .
C.Time -> Day.Time | Year.Time
Day.Time -> Morning | Afternoon | Evening | Night | . . .
Year.Time -> Winter | Spring | Summer | Fall
Weather -> Sunny | Rain | Snow
Direction -> Straight | Steer
Steer -> Left | Right | Hard | Gently
Left -> Left&Hard | Left&Gently
Right -> Right&Hard | Right&Gently
Hard -> Left&Hard | Right&Hard
Gently -> Left&Gently | Right&Gently
Speed -> Constant | Accelerate | Break
Constant -> Slow | Medium | Fast
Accelerate -> A.Hard | A.Slowly
Break -> B.Hard | B.Slowly
Adjust_Control -> Window | Radio | . . .
Sound -> Type | Intensity
Type -> Gentle | Harsh | . . .
Intensity -> Soft | Loud | . . . }

S = Situation

Most of the driving domain situations encountered during this thesis can now be obtained through a number of different derivations in this

grammar. Also, many other situations in the driving domain can be expressed using this vocabulary. Clearly, this vocabulary is not enough to describe all possible contingencies in the driving domain. It was not our goal to provide such a vocabulary and grammar. However, it can be easily extended to encompass, in the same domain, any other desired situation which cannot be represented yet.

Contingencies and reactions are, in general, associated with sets of situations. Therefore, general situations are most often enough to be specified, and the derivations may be stopped at those levels where the situation expressed by the sentential form obtained thus far "contains" (according to the order relation defined in chapter 4) all the elementary situations to which the contingency or reaction apply. This knowledge structuring property of the representation formalism is most important here, since it helps contain the explosion of the situations in the domain, ensuring the representability of the knowledge needed for our planning-to-react decision framework in large domains.

While most situations encountered in chapter 3 can be derived in this formalism, it also supports the derivation of many other situations for the driving domain. In fact, just by enlarging the set of terminals, the number of situations expressible with this small grammar becomes very large indeed. This fact underlines the most important advantage of this representation formalism, namely imposing a (hierarchical) structure on the set of possible situations in the domain, which then makes them much easier to be stored, managed, analyzed and reasoned about.

Appendix 3
Anesthesiology Domain Experiments

In order to demonstrate the applicability and scalability of the reaction decision framework presented in chapter 3, we have run demonstrations in one other domain than those described in chapter 6. We briefly describe here these demonstrations. The domain is anesthesiology, and I am indebted to Dr. David Gaba for letting me benefit from his time and knowledge by serving the role of the domain expert both for the knowledge acquisition task, as well as for the evaluation phase of the experiments. Working in a professional domain of high expertise, we have used this time a single expert to provide us the necessary knowledge (in contrast with the driving domain where we have acquired it through a statistical analysis of the opinions of a group of experts in the domain, as explained in section 6.1).

Table A3.1 lists the set of 13 contingencies selected for this experiment, together with the reactions for each of them (in the "random" order specified by the expert), for the following situation:

Problem:	Anesthetize patient for bowel obstruction
Plan:	Induce anesthesia [rapid sequence induction]
Context:	Middle of the night, emergency case, patient has coronary artery disease (moderate) and chronic obstructive pulmonary disease (severe)
Ext. Expect.:	Change in vital signs
Int. Expect.:	Patient becomes unresponsive to commands

Action: Rapid sequence induction (Pentothal and Succinylcholinc have just been administered)

$Time_S$: 60 seconds.

The expert was asked to translate his qualitative feelings into quantitative values, and to concentrate more on relative values than on the absolute values he was going to specify. The expert was not asked to order the contingencies as he feels would be appropriate for a normal behavior. Rather, we have presented him with the system's results and ask him to evaluate the behavior recommended by our framework. The knowledge acquired from the expert was for the following contingency characteristics: time to respond (real values in seconds), criticality, side-effects, and likelihood (all these three on a scale of [0,10]).

	Contingency	Reaction
1	Patient vomits	Turn head; suction mouth; intubate
2	Patient does not "fall asleep"	Check IV and syringe; give more drug
3	Muscle fasciculations (twitching 2" to drug)	Ensure patient does not fall asleep
4	Decreased blood pressure	Increase IV rate; administer vasopressor
5	Increased heart rate	Consider deeper anesthesia or β blocker
6	Cardiac Arrest	ACLS (Advanced Cardiac Life Support)
7	Meteor strikes OR	Move patient out of OR
8	Failure of pipeline oxygen supply	Switch tanks ON; disconnect pipeline
9	Failure of 1° and backup electric power	Obtain flashlight
10	Inability to intubate trachea	Ventilate by mask if possible; emergency procedures for difficult airway
11	Message from PACU about previous patient	Listen to the message
12	Severe bronchospasm (wheezing)	Ensure correct intubation; treat with bronchodilators
13	O2 saturation decreases to < 90%	Ventilate by mask or tube with 100% O2

Table A3.1. Contingencies for the anesthesia domain experiments

We have also asked the expert to calibrate his data by supplying values for the expert model parameters for the recommended behavior model. These values were: 1 second for minimum real time (corresponding to T_{max}), 30 minutes for maximum real time threshold (corresponding to T_{min}), 1.0 for

minimum likelihood (L_{min}), and 2.3 for the difference between consequences and side-effects (CS_{min}). We did not ask the expert to actually give us a function to translate from real time to time pressure, but rather we have specified it ourselves, in such a way as to include most of the time pressures in the interval [0,10]. The function we came up with is:

$$f_{tc} = k / time_{rc} = 50 / time_{rc} .$$

We have experimented with significantly different values for k (between [10,100]) and the results obtained were remarkably similar (actually most were identical) with the ones reported here. However, we have settled for the value k = 50, for the reason stated above (all but one time pressure values are between [0,10], with a reasonable spread in this interval). The results of the knowledge acquisition process in this domain are summarized in table A3.2 (in the same order as the previous table).

	Contingency	$time_{rc}$	$time_p$	consequence	side-effect	likelihood
1	vomit	15.0	3.33	8.0	2.0	7.0
2	not fall asleep	45.0	1.11	7.0	4.0	4.0
3	muscle fascic.	100.0	0.5	3.0	1.0	8.0
4	decreased BP	15.0	3.33	8.0	5.0	6.0
5	increased HR	15.0	3.33	6.0	6.0	7.0
6	cardiac arrest	5.0	10.0	10.0	7.0	2.0
7	meteor	0.1	500.0	9.0	7.0	0.01
8	O2 supply fails	30.0	1.67	8.5	5.0	1.0
9	power failure	30.0	1.67	6.0	5.0	1.0
10	can't intubate	10.0	5.0	9.5	8.0	5.0
11	PACU message	200.0	0.25	1.0	1.0	4.0
12	bronchospasm	25.0	2.0	9.0	7.0	6.0
13	O2 sat < 90%	15.0	3.33	8.0	4.0	6.0

Table A3.2. Data values for the anesthesiology domain experiments

We have first run the "normal" behavior model on these contingencies. The values for the criticality function parameters given by the behavior model were the same as for the driving domain:

$$p_1 = 5, p_2 = 1, p_3 = 0, p_4 = 0, p_5 = 3, p_6 = 2,$$

with the parameters specified by the expert model (and discussed above) close to those given in section 6.2. Table A3.3 summarizes the results of this run. The contingencies are this time numbered in the order specified by the criticality

function for this case, which we shall call from now on the "system-recommended" order (since it was obtained by running the system with the recommended behavior model). There are two possible monitoring thresholds, since there are two significant gaps in the sequence of values returned by the criticality function.

Contingency		Criticality	Monitor
1	cardiac arrest	5.95E8	yes
2	vomit	9.22E7	yes
3	can't intubate	4.07E7	yes
4	O2 sat < 90%	2.96E7	yes
5	decreased BP	1.76E7	yes
6	increased HR	1.47E6	yes
7	bronchospasm	8.24E5	yes
8	not fall asleep	2.82E4	??
9	O2 supply fail	2.13E4	??
10	power failure	2.77E3	??
11	muscle fascic.	4.77E2	??
12	PACU messg	0.19	
13	meteor	0.00	

Table A3.3. Criticality values for the "normal" behavior model, for the anesthesiology domain experiments

As mentioned before, the expert was not required to order the contingencies by reaction value according to his belief of what the recommended behavior should be like. However, when presented with the results, he characterized them as "definitely reasonable". This shows a significant portability of the behavior model and of all the parameter values for the criticality function, across domains (since the driving and anesthesiology domains are significantly different in nature, and the experts have specified their knowledge in the two domains independent of each other).

We have then run our framework on this data, for all the other behavior models defined in section 6.3. We summarize in table A3.4 the values we have used for the criticality function parameters in each run for this domain. Note that all the behavior model parameters (p_1 to p_6) have received identical values for the two domains. Also most of the expert model parameters are unchanged, and the changes reflect the different calibrations of the experts when they have specified the data.

Behavior	Behavior Model p_1	p_2	p_3	p_4	p_5	p_6	Expert Model T_{max}	T_{min}	CS_{max}	L_{min}
Recommended	5	1	0	0	3	2	50.0	0.028	2.3	1.0
Antiauthority	5	1	0	0	3	0				
Impulsivity	0	0	0	0	0	3		5.0		5.0
Invulnerability	5	1	0	0	3	2				5.2
Macho	4	1	0	0	0	3			10.0	
Resignation	5	1	0	0	3	2	2.0			
Risk-averse	2	2	4	2	1	1				
Liability conscious	3	3	1	2	1	2	500.0	0.0		0.0
Social responsibility	4	3	0	0	4	3				

Table A3.4 Representing Behavior Models

Table A3.5 summarizes the results of these experiments. We have also shown the reaction values produced by the criticality function. Their absolute values have no meaning whatsoever; what matters are their relative values (and only within the same behavior model), which represent the relative value of reacting to one contingency vs. another in a same situation. For each behavior, monitoring thresholds were set (for the expert model) in regions of the contingency space where there are big gaps among the reaction values of the contingencies ordered by criticality. The thresholds are represented by thicker lines separating the contingencies for each behavior into two or three sets (in many cases, two possible places were indicated for this threshold).

Behavior Model 1 (Recommended)			Behavior Model 2 (Antiauthority)			Behavior Model 3 (Impulsivity)		
1	cardiac arrest	5.95E8	1	cardiac arrest	1.48E8	3	can't intubate	1.25E2
2	vomit	9.22E7	2	vomit	1.88E6	11	muscle fascic.	22.62
3	can't intubate	4.07E7	3	can't intubate	1.62E6	2	vomit	18.52
4	O2 sat < 90%	2.96E7	4	O2 sat < 90%	8.23E5	6	increased HR	18.52
6	decreased BP	1.76E7	5	decreased BP	4.90E5	7	bronchospasm	14.69
5	increased HR	1.47E6	6	increased HR	3.00E4	5	decreased BP	14.69
7	bronchospasm	8.24E5	7	bronchospasm	2.28E4	4	O2 sat < 90%	14.69
8	not fall asleep	2.82E4	9	O2 supply fail	2.13E4	8	not fall asleep	8.00
9	O2 supply fail	2.13E4	10	power failure	2.77E3	12	PACU messg	8.00
10	power failure	2.77E3	8	not fall asleep	1.76E3	1	cardiac arrest	2.82
11	muscle fascic.	4.77E2	11	muscle fascic.	7.45	9	O2 supply fail	1.00
12	PACU messg	0.19	12	PACU messg	1.1E-2	10	power failure	1.00
13	meteor	0.00	13	meteor	0.00	13	meteor	0.00

Table A3.5 Reactive Behavior Experiments for Anesthesiology

Behavior Model 4 (Invulnerability)			Behavior Model 5 (Macho)			Behavior Model 6 (Resignation)		
2	vomit	9.22E7	1	cardiac arrest	8.00E5	7	bronchospasm	8.24E5
4	O2 sat < 90%	2.96E7	3	can't intubate	7.42E5	8	not fall asleep	2.82E4
5	decreased BP	1.76E7	2	vomit	3.28E5	9	O2 supply fail	2.13E4
6	increased HR	1.47E6	6	increased HR	2.54E5	10	power failure	2.77E3
7	bronchospasm	8.24E5	5	decreased BP	2.13E5	11	muscle fascic.	4.77E2
1	cardiac arrest	2.44E4	4	O2 sat < 90%	2.13E5	12	PACU messg	0.19
3	can't intubate	6.38E3	7	bronchospasm	3.11E4	1	cardiac arrest	0.00
11	muscle fascic.	4.77E2	8	not fall asleep	6.82E2	3	can't intubate	0.00
8	not fall asleep	1.68E2	11	muscle fascic.	96.00	13	meteor	0.00
9	O2 supply fail	1.46E2	9	O2 supply fail	65.58	2	vomit	0.00
10	power failure	52.65	10	power failure	46.29	5	decreased BP	0.00
12	PACU messg	0.43	12	PACU messg	0.25	4	O2 sat < 90%	0.00
13	meteor	0.00	13	meteor	0.00	6	increased HR	0.00

Table A3.5 Reactive Behavior Experiments for Anesthesiology (continued)

Behavior Model 7 (Risk-averse)			Behavior Model 8 (Liability conscious)			Behavior Model 9 (Social responsibility)		
1	cardiac arrest	7.3E10	13	meteor	7.0E10	2	vomit	1.0E11
3	can't intubate	5.3E10	1	cardiac arrest	4.2E10	1	cardiac arrest	6.3E10
7	bronchospasm	5.13E9	3	can't intubate	2.4E10	4	O2 sat < 90%	2.1E10
5	decreased BP	2.38E9	5	decreased BP	3.05E9	3	can't intubate	1.3E10
6	increased HR	1.20E9	4	O2 sat < 90%	2.47E9	5	decreased BP	1.0E10
4	O2 sat < 90%	9.90E8	7	bronchospasm	1.61E9	7	bronchospasm	8.61E8
9	O2 supply fail	1.32E8	2	vomit	1.54E9	6	increased HR	2.55E8
2	vomit	6.61E7	6	increased HR	7.78E8	8	not fall asleep	2.64E7
8	not fall asleep	3.97E7	8	not fall asleep	1.93E7	9	O2 supply fail	5.36E6
10	power failure	2.49E7	9	O2 supply fail	1.50E7	10	power failure	1.97E5
11	muscle fascic.	1.23E3	10	power failure	1.99E6	11	muscle fascic.	2.95E5
12	PACU messg	2.30	11	muscle fascic.	1.48E4	12	PACU messg	6.99
13	meteor	0.00	12	PACU messg	2.30	13	meteor	0.00

Table A3.5 Reactive Behavior Experiments for Anesthesiology (continued)

The numbering of contingencies for each behavior model in table A3.5 is the same as for the recommended behavior. This was done in order to facilitate comparisons of each behavior model with the "normal" one.

In chapter 5, we have defined a behavior model to be an order relationship on the set of contingencies associated with a situation. Therefore, in these experiments, we only concentrate on the ordering of contingencies by reaction value (and sometimes relative values of the criticality function,

but never on its absolute values), and ignore any issues related to the reactive planner model and the agent model, that is we ignore the final decision of applying the framework to a set of contingencies. This is consistent with the purpose of our demonstrations here, since any specific agent (with a given reactive planner and resource limitations) may exhibit any of the reaction behaviors discussed, depending only on the order in which its behavior model recommends the contingencies for consideration to be reacted to, and not on the actual components and resources of the agent.

The results of these demonstrations require a certain amount of interpretation (this is necessary especially since the definitions of these behavior models are generally based on execution time types of reactions, while we attempt here to implement them at planning time). For example, for the *antiauthority* behavior model, the order of contingencies does not change much, since here almost all contingencies considered are covered by regulations or procedures; only "not fall asleep" goes down since after all this is precisely what we want to achieve and is therefore best covered by procedures in this case. In the *invulnerability* case, "cardiac arrest" and "can't intubate" fall significantly (possibly even below the monitoring threshold) because they are not likely enough in this particular situation (for this particular patient) where the likelihood threshold has been increased due to the type of behavior under consideration. Also "muscle fasciculations" advances a lot because of its high likelihood compared to the other contingencies. In the *liability conscious* behavior, the agent considers almost all consequences, except "message from PACU" because of its very long time of response which should allow for replanning (here "meteor strikes operating room" becomes very high priority, since once it is considered - regardless of its much too short response time allowed - its very high time pressure and consequences make it very high priority. Similar arguments can be made for the results of each of the behavior models used in this demonstration.

The interpretation of our results shows (in the expert's opinion) that they are reasonable and consistent with the generally accepted (execution-time) definition of each behavior model, and that there is a plausible explanation for the results that maps them into the corresponding (conceptual) behaviors. These demonstrations again show that our formalism may at least provide a reasonable basis for representing and exchanging

information and ideas about reaction-related behavior models, and thus for interpreting and studying different behaviors, in a considerable variety of domains (from mundane tasks like car driving, to highly specialized ones like medical domains). A possible use is to start from a specific behavior (order on the set of contingencies) exhibited by an agent, discover - using machine learning techniques - the parameters of the behavior model which emulates this behavior in our framework, and then use these parameters to characterize the behavior and maybe to attempt to consciously modify it. However, these are only speculations at this point, since as stated before, much research is still needed to refine such a behavior description formalism into a useful tool for changing ideas among behavioral experts.

Appendix 4
Intensive Care Domain Experiments

We present here some of the results of the experiments we have conducted with our framework in the intensive care monitoring domain. This appendix mainly complements section 6.4.

#	Contingency (Response would be the typical response for this event)	Response time (min)	Conse-quences	Side-effects	Likeli-hood
1	myocardial-depression-post-cpb	10	8.5	7	3
2	myocardial-depression-sepsis	20	8	7.5	1
3	decreased-preload	20	7	3	7
4	increased-afterload	20	6.5	5	4
5	cardiac-tamponade	5	8.5	7.5	3
6	hypovolemia	20	7	3	7
7	myocardial-ischemia	5	8	6	3
8	myocardial-infarction	60	6	5	3
9	right-heart-failure	10	8	7	2
10	digitalis-toxicity	180	5	4	2
11	postop-hypertension	20	6.5	5	4
12	cardiac-arrest	1	10	8	1
13	ventricular-fibrillation	1	10	8	1
14	ventricular-ectopy	5	7	7	6
15	sinus-bradycardia	5	7	5	3
16	atrial-fibrillation	20	7	6	4
17	paroxysmal-supraventric-tachycardia	20	6	6	4
18	ventricular-tachycardia	1	9	7	2
19	sinus-tachycardia	10	6	5	7
20	hypoxia	5	8	6	4
21	respiratory-acidosis	60	6	4	4
22	cardiogenic-pulmonary-edema	10	8.5	7	3
23	noncardiogenic-pulmonary-edema	20	8.5	8	2

Table A4.1 Contingencies for the ICU domain

24	atelectasis	120	6.5	5	6.5
25	pneumothorax	10	8	7	3
26	hemothorax	10	7	7	4
27	chylothorax	120	7	7	2
28	aspiration-pneumonia	240	8	5	1
29	pneumonia	240	7	5	3
30	diaphragmatic-paralysis	600	8	7	1
31	bronchospasm	30	8	7	4
32	pulmonary-embolism	10	8.5	7.5	3
33	ARDS	120	8.5	8	2
34	et-tube-disconnection	2	10	2	4
35	kinked-et-tube	5	8	2	4
36	right-mainstem-intubation	20	6.5	3	2
37	disseminated-intravascular-coagulat	60	8	7	2
38	dilutional-coagulopathy	60	7	3	5
39	platelet-deficiency	60	7	3	5
40	acute-hemolytic-transfusion-react	10	8.5	5	1
41	febrile-nonhemolytic-transfus-react	20	6.5	4	2
42	mechanical-bleeding	20	7.5	7.5	4
43	fibrinogen-defects	60	7	3	5
44	extrinsic-pathway-defects	60	7	3	5
45	intrinsic-pathway-defects	60	7	3	5
46	cerebrovascular-ischemia	60	8.5	7.5	2
47	cerebrovascular-embolism	30	9	7.5	1
48	endotoxemia	120	8.5	8	1
49	rewarming	240	3	3	7
50	hypothermia	240	4	4	7
51	hyperglycemia	120	5	4	2
52	metabolic-acidosis	60	6.5	4	3
53	acute-renal-failure	300	9	8	1
54	acute-tubular-necrosis	300	9	8	1
55	prerenal-azotemia	300	5	5	3
56	renal-azotemia	300	5	6	1
57	renal-embolism	300	7	7	1
58	high-cl	120	6	4	6
59	low-cl	120	6	4	2
60	high-ca	60	7	6	1
61	low-ca	60	6	3	6
62	low-mg	60	7	3	7
63	high-mg	60	8	5	2
64	low-na	30	7	2	2
65	high-na	60	6	3	2
66	dilutional-low-na	30	7	2	2
67	low-k	30	7.5	5	5
68	high-k	30	8	7	4

Table A4.1 Contingencies for the ICU domain (continued)

#	Contingency (Response would be the typical response for this event)	Resp. time	Conse-quences	Side-eff.	Likeli-hood	Criti-cality
34	et-tube-disconnection	2	10	2	4	4.2E12
18	ventricular-tachycardia	1	9	7	2	2.2E12
13	ventricular-fibrillation	1	10	8	1	6.1E11
12	cardiac-arrest	1	10	8	1	6.1E11
35	kinked-et-tube	5	8	2	4	1.8E10
20	hypoxia	5	8	6	4	2.53E9
7	myocardial-ischemia	5	8	6	3	1.42E9
15	sinus-bradycardia	5	7	5	3	1.24E9
14	ventricular-ectopy	5	7	7	6	7.62E8
5	cardiac-tamponade	5	8.5	7.5	3	6.84E8
19	sinus-tachycardia	10	6	5	7	8.21E7
22	cardiogenic-pulmonary-edema	10	8.5	7	3	3.26E7
1	myocardial-depression-post-cpb	10	8.5	7	3	3.26E7
32	pulmonary-embolism	10	8.5	7.5	3	2.13E7
6	hypovolemia	20	7	3	7	2.08E7
3	decreased-preload	20	7	3	7	2.08E7
25	pneumothorax	10	8	7	3	2.01E7
40	acute-hemolytic-transfusion-react	10	8.5	5	1	1.28E7
26	hemothorax	10	7	7	4	1.05E7
9	right-heart-failure	10	8	7	2	8.94E6
11	postop-hypertension	20	6.5	5	4	1.38E6
4	increased-afterload	20	6.5	5	4	1.38E6
36	right-mainstem-intubation	20	6.5	3	2	1.23E6
16	atrial-fibrillation	20	7	6	4	9.78E5
41	febrile-nonhemolytic-transfus-react	20	6.5	4	2	6.98E5
67	low-k	30	7.5	5	5	6.63E5
42	mechanical-bleeding	20	7.5	7.5	4	3.54E5
66	dilutional-low-na	30	7	2	2	3.48E5
64	low-na	30	7	2	2	3.48E5
17	paroxysmal-supraventric-tachycardia	20	6	6	4	2.83E5
23	noncardiogenic-pulmonary-edema	20	8.5	8	2	1.81E5
68	high-k	30	8	7	4	1.47E5
31	bronchospasm	30	8	7	4	1.47E5
62	low-mg	60	7	3	7	8.57E4
45	intrinsic-pathway-defects	60	7	3	5	4.37E4
44	extrinsic-pathway-defects	60	7	3	5	4.37E4
43	fibrinogen-defects	60	7	3	5	4.37E4
39	platelet-deficiency	60	7	3	5	4.37E4
38	dilutional-coagulopathy	60	7	3	5	4.37E4
2	myocardial-depression-sepsis	20	8	7.5	1	4.26E4
61	low-ca	60	6	3	6	3.21E4
47	cerebrovascular-embolism	30	9	7.5	1	1.58E4
21	respiratory-acidosis	60	6	4	4	7.63E3

Table A4.2. ICU domain contingencies ordered by criticality for $T_{min} = 0.5$ (2 hours) and $L_{min} = 1$

52	metabolic-acidosis	60	6.5	4	3	6.46E3
63	high-mg	60	8	5	2	4.76E3
65	high-na	60	6	3	2	3.57E3
8	myocardial-infarction	60	6	5	3	1.94E3
46	cerebrovascular-ischemia	60	8.5	7.5	2	1.22E3
37	disseminated-intravascular-coagulat	60	8	7	2	1.14E3
58	high-cl	120	6	4	6	5.36E2
24	atelectasis	120	6.5	5	6.5	4.70E2
60	high-ca	60	7	6	1	2.51E2
59	low-cl	120	6	4	2	59.63
33	ARDS	120	8.5	8	2	23.32
51	hyperglycemia	120	5	4	2	22.46
27	chylothorax	120	7	7	2	10.64
48	endotoxemia	120	8.5	8	1	5.83
29	pneumonia	240	7	5	3	2.21
10	digitalis-toxicity	180	5	4	2	1.71
50	hypothermia	240	4	4	7	1.52
49	rewarming	240	3	3	7	1.32
28	aspiration-pneumonia	240	8	5	1	1.07
55	prerenal-azotemia	300	5	5	3	0.41
54	acute-tubular-necrosis	300	9	8	1	0.32
53	acute-renal-failure	300	9	8	1	0.32
57	renal-embolism	300	7	7	1	0.16
56	renal-azotemia	300	5	6	1	5.9E-2
30	diaphragmatic-paralysis	600	8	7	1	5.3E-2

Table A4.2. ICU domain contingencies ordered by criticality
for T_{min} = 0.5 (2 hours) and L_{min} = 1 (continued)

Table A4.1 lists the entire set of 68 contingencies defined by the experts in the domain for the situations described in figure 6.1, together with their characteristic valuesm. The contingencies are listed in the order specified by the experts (grouped by categories of complications that may develop).

The first part of this demonstration consisted in running the criticality function part of the framework on this data set, for the recommended behavior model (section 6.3). We have done this for several expert models which differ in the minimum time pressure threshold (T_{min}) value, and the minimum likelihood threshold (L_{min}) value. We shall present here only the results of four such experiments, although we have made a much larger number.

Table A4.2 shows the order of the contingencies given by the "normal" behavior model for a maximum reaction time of 2 hours (T_{min} = 0.5) and a

minimum likelihood of 1. The rest of the expert model parameters are left unchanged during all these experiments (they are: $f_{tc} = 60 / time_{rc}$; $T_{max} = 100$ (36 seconds); $CS_{min} = 2.3$).

#	Contingency (Response would be the typical response for this event)	Resp. time	Conse-quences	Side-eff.	Likeli-hood	Criti-cality
34	et-tube-disconnection	2	10	2	4	4.2E12
18	ventricular-tachycardia	1	9	7	2	2.2E12
35	kinked-et-tube	5	8	2	4	1.8E10
20	hypoxia	5	8	6	4	2.53E9
7	myocardial-ischemia	5	8	6	3	1.42E9
15	sinus-bradycardia	5	7	5	3	1.24E9
14	ventricular-ectopy	5	7	7	6	7.62E8
5	cardiac-tamponade	5	8.5	7.5	3	6.84E8
19	sinus-tachycardia	10	6	5	7	8.21E7
22	cardiogenic-pulmonary-edema	10	8.5	7	3	3.26E7
1	myocardial-depression-post-cpb	10	8.5	7	3	3.26E7
32	pulmonary-embolism	10	8.5	7.5	3	2.13E7
6	hypovolemia	20	7	3	7	2.08E7
3	decreased-preload	20	7	3	7	2.08E7
25	pneumothorax	10	8	7	3	2.01E7
26	hemothorax	10	7	7	4	1.05E7
9	right-heart-failure	10	8	7	2	8.94E6
11	postop-hypertension	20	6.5	5	4	1.38E6
4	increased-afterload	20	6.5	5	4	1.38E6
36	right-mainstem-intubation	20	6.5	3	2	1.23E6
16	atrial-fibrillation	20	7	6	4	9.78E5
13	ventricular-fibrillation	1	10	8	1	7.86E5
12	cardiac-arrest	1	10	8	1	7.86E5
41	febrile-nonhemolytic-transfus-react	20	6.5	4	2	6.98E5
67	low-k	30	7.5	5	5	6.63E5
42	mechanical-bleeding	20	7.5	7.5	4	3.54E5
66	dilutional-low-na	30	7	2	2	3.48E5
64	low-na	30	7	2	2	3.48E5
17	paroxysmal-supraventric-tachycardia	20	6	6	4	2.83E5
23	noncardiogenic-pulmonary-edema	20	8.5	8	2	1.81E5
68	high-k	30	8	7	4	1.47E5
31	bronchospasm	30	8	7	4	1.47E5
62	low-mg	60	7	3	7	8.57E4
45	intrinsic-pathway-defects	60	7	3	5	4.37E4
44	extrinsic-pathway-defects	60	7	3	5	4.37E4
43	fibrinogen-defects	60	7	3	5	4.37E4
39	platelet-deficiency	60	7	3	5	4.37E4
38	dilutional-coagulopathy	60	7	3	5	4.37E4
61	low-ca	60	6	3	6	3.21E4
21	respiratory-acidosis	60	6	4	4	7.63E3

Table A4.3. ICU domain contingencies ordered by criticality

for T_{min} = 0.5 (2 hours) and L_{min} = 2

52	metabolic-acidosis	60	6.5	4	3	6.46E3
63	high-mg	60	8	5	2	4.76E3
40	acute-hemolytic-transfusion-react	10	8.5	5	1	3.59E3
65	high-na	60	6	3	2	3.57E3
8	myocardial-infarction	60	6	5	3	1.94E3
46	cerebrovascular-ischemia	60	8.5	7.5	2	1.22E3
37	disseminated-intravascular-coagulat	60	8	7	2	1.14E3
58	high-cl	120	6	4	6	5.36E2
24	atelectasis	120	6.5	5	6.5	4.70E2
2	myocardial-depression-sepsis	20	8	7.5	1	2.06E2
47	cerebrovascular-embolism	30	9	7.5	1	1.25E2
59	low-cl	120	6	4	2	59.63
33	ARDS	120	8.5	8	2	23.32
51	hyperglycemia	120	5	4	2	22.46
60	high-ca	60	7	6	1	15.86
27	chylothorax	120	7	7	2	10.64
48	endotoxemia	120	8.5	8	1	2.41
29	pneumonia	240	7	5	3	2.21
10	digitalis-toxicity	180	5	4	2	1.71
50	hypothermia	240	4	4	7	1.52
49	rewarming	240	3	3	7	1.32
28	aspiration-pneumonia	240	8	5	1	1.07
55	prerenal-azotemia	300	5	5	3	0.41
54	acute-tubular-necrosis	300	9	8	1	0.32
53	acute-renal-failure	300	9	8	1	0.32
57	renal-embolism	300	7	7	1	0.16
56	renal-azotemia	300	5	6	1	5.9E-2
30	diaphragmatic-paralysis	600	8	7	1	5.3E-2

Table A4.3. ICU domain contingencies ordered by criticality for T_{min} = 0.5 (2 hours) and L_{min} = 2 (continued)

To show the effect of varying the likelihood parameter in the expert model, table A4.3 presents the ordering of contingencies according to the same behavior model, with all the parameters unchanged except the minimum likelihood raised at 2. We can see that highly consequential but low likelihood contingencies like *ventricular-fibrillation* and *cardiac-arrest* experience a significant drop in criticality (from the 3rd place to the 22nd). However, their high consequences and high time pressure ensure that they do not fall very much (they are still ranked by the framework in the first third of all the contingencies considered).

#	Contingency (Response would be the typical response for this event)	Resp. time	Consequences	Side-eff.	Likelihood	Criticality
34	et-tube-disconnection	2	10	2	4	4.2E12
18	ventricular-tachycardia	1	9	7	2	2.2E12
13	ventricular-fibrillation	1	10	8	1	6.1E11
12	cardiac-arrest	1	10	8	1	6.1E11
35	kinked-et-tube	5	8	2	4	1.8E10
20	hypoxia	5	8	6	4	2.53E9
7	myocardial-ischemia	5	8	6	3	1.42E9
15	sinus-bradycardia	5	7	5	3	1.24E9
14	ventricular-ectopy	5	7	7	6	7.62E8
5	cardiac-tamponade	5	8.5	7.5	3	6.84E8
19	sinus-tachycardia	10	6	5	7	8.21E7
22	cardiogenic-pulmonary-edema	10	8.5	7	3	3.26E7
1	myocardial-depression-post-cpb	10	8.5	7	3	3.26E7
32	pulmonary-embolism	10	8.5	7.5	3	2.13E7
6	hypovolemia	20	7	3	7	2.08E7
3	decreased-preload	20	7	3	7	2.08E7
25	pneumothorax	10	8	7	3	2.01E7
40	acute-hemolytic-transfusion-react	10	8.5	5	1	1.28E7
26	hemothorax	10	7	7	4	1.05E7
9	right-heart-failure	10	8	7	2	8.94E6
11	postop-hypertension	20	6.5	5	4	1.38E6
4	increased-afterload	20	6.5	5	4	1.38E6
36	right-mainstem-intubation	20	6.5	3	2	1.23E6
16	atrial-fibrillation	20	7	6	4	9.78E5
41	febrile-nonhemolytic-transfus-react	20	6.5	4	2	6.98E5
67	low-k	30	7.5	5	5	6.63E5
42	mechanical-bleeding	20	7.5	7.5	4	3.54E5
66	dilutional-low-na	30	7	2	2	3.48E5
64	low-na	30	7	2	2	3.48E5
17	paroxysmal-supraventric-tachycardia	20	6	6	4	2.83E5
23	noncardiogenic-pulmonary-edema	20	8.5	8	2	1.81E5
68	high-k	30	8	7	4	1.47E5
31	bronchospasm	30	8	7	4	1.47E5
2	myocardial-depression-sepsis	20	8	7.5	1	4.26E4
47	cerebrovascular-embolism	30	9	7.5	1	1.58E4
62	low-mg	60	7	3	7	2.92E2
45	intrinsic-pathway-defects	60	7	3	5	2.09E2
44	extrinsic-pathway-defects	60	7	3	5	2.09E2
43	fibrinogen-defects	60	7	3	5	2.09E2
39	platelet-deficiency	60	7	3	5	2.09E2
38	dilutional-coagulopathy	60	7	3	5	2.09E2
61	low-ca	60	6	3	6	1.79E2
21	respiratory-acidosis	60	6	4	4	87.36

Table A4.4. ICU domain contingencies ordered by criticality for T_{min} = 2 (30 minutes) and L_{min} = 1

52	metabolic-acidosis	60	6.5	4	3	80.43
63	high-mg	60	8	5	2	69.02
65	high-na	60	6	3	2	59.77
8	myocardial-infarction	60	6	5	3	44.05
46	cerebrovascular-ischemia	60	8.5	7.5	2	34.95
37	disseminated-intravascular-coagulat	60	8	7	2	33.91
58	high-cl	120	6	4	6	23.16
24	atelectasis	120	6.5	5	6.5	21.70
60	high-ca	60	7	6	1	15.86
59	low-cl	120	6	4	2	7.72
33	ARDS	120	8.5	8	2	4.82
51	hyperglycemia	120	5	4	2	4.73
27	chylothorax	120	7	7	2	3.26
48	endotoxemia	120	8.5	8	1	2.41
29	pneumonia	240	7	5	3	2.21
10	digitalis-toxicity	180	5	4	2	1.71
50	hypothermia	240	4	4	7	1.52
49	rewarming	240	3	3	7	1.32
28	aspiration-pneumonia	240	8	5	1	1.07
55	prerenal-azotemia	300	5	5	3	0.41
54	acute-tubular-necrosis	300	9	8	1	0.32
53	acute-renal-failure	300	9	8	1	0.32
57	renal-embolism	300	7	7	1	0.16
56	renal-azotemia	300	5	6	1	5.9E-2
30	diaphragmatic-paralysis	600	8	7	1	5.3E-2

Table A4.4. ICU domain contingencies ordered by criticality for T_{min} = 2 (30 minutes) and L_{min} = 1 (continued)

Tables A4.4 and A4.5 show the effect of increasing the time pressure threshold. While table A4.2 contains the contingencies ordered according to an expert model which recommends reactions for contingencies with allowed response time of up to 2 hours from the time a contingency is detected, table A4.4 reduces this time to half an hour (minimum time pressure T_{min} = 2), and table A4.5 reduces it even further, to just 5 minutes (minimum time pressure T_{min} = 12). Notice that contingencies with very low likelihood but higher time pressure (like *myocardial-depression-sepsis* and *cerebrovascular-embolism*) advance over more likely contingencies but with time pressure lower than the recommended reaction threshold, in table A4.4. However, when the time pressure threshold is raised significantly more (table A4.5), we obtain an identical ordering with the initial one in table A4.2, because the expert has recommended reactions only for very time critical contingencies, which were ranked as having high criticality by the framework even from the beginning,

other things being equal. There is however a significant difference between tables A4.2 and A4.5 (and to a lesser extent table A4.4), namely a clear threshold for monitoring. In the case of a very low time pressure threshold (2 hours), there is no such clear threshold, since the criticality of contingencies decreases gradually in table A4.2, without a clear gap. This is because, when the maximum reaction time recommended is very large, the time pressure for contingencies with long allowed response time is so small anyway, that it does not influence the criticality of that contingency too much. This contrasts with the cases when the maximum reaction time recommended is small, for which the time pressure is high enough to make a significant difference in the criticality value. This is why in table A4.5 we have a clear threshold (given by a significant gap in the sequence of criticality values) after the 10th contingency in the sequence (*cardiac-tamponade*). The same phenomenon takes place in table A4.4 after the *cerebrovascular-embolism* contingency.

#	Contingency (Response would be the typical response for this event)	Resp. time	Conse-quences	Side-eff.	Likeli-hood	Criti-cality
34	et-tube-disconnection	2	10	2	4	4.2E12
18	ventricular-tachycardia	1	9	7	2	2.2E12
13	ventricular-fibrillation	1	10	8	1	6.1E11
12	cardiac-arrest	1	10	8	1	6.1E11
35	kinked-et-tube	5	8	2	4	1.8E10
20	hypoxia	5	8	6	4	2.53E9
7	myocardial-ischemia	5	8	6	3	1.42E9
15	sinus-bradycardia	5	7	5	3	1.24E9
14	ventricular-ectopy	5	7	7	6	7.62E8
5	cardiac-tamponade	5	8.5	7.5	3	6.84E8
19	sinus-tachycardia	10	6	5	7	9.06E3
22	cardiogenic-pulmonary-edema	10	8.5	7	3	5.71E3
1	myocardial-depression-post-cpb	10	8.5	7	3	5.71E3
32	pulmonary-embolism	10	8.5	7.5	3	4.62E3
6	hypovolemia	20	7	3	7	4.56E3
3	decreased-preload	20	7	3	7	4.56E3
25	pneumothorax	10	8	7	3	4.48E3
40	acute-hemolytic-transfusion-react	10	8.5	5	1	3.59E3
26	hemothorax	10	7	7	4	3.25E3
9	right-heart-failure	10	8	7	2	2.99E3
11	postop-hypertension	20	6.5	5	4	1.17E3
4	increased-afterload	20	6.5	5	4	1.17E3
36	right-mainstem-intubation	20	6.5	3	2	1.11E3

Table A4.5. ICU domain contingencies ordered by criticality
for T_{min} = 12 (5 minutes) and L_{min} = 1

16	atrial-fibrillation	20	7	6	4	9.88E2
41	febrile-nonhemolytic-transfus-react	20	6.5	4	2	8.35E2
67	low-k	30	7.5	5	5	8.14E2
42	mechanical-bleeding	20	7.5	7.5	4	5.95E2
66	dilutional-low-na	30	7	2	2	5.90E2
64	low-na	30	7	2	2	5.90E2
17	paroxysmal-supraventric-tachycardia	20	6	6	4	5.32E2
23	noncardiogenic-pulmonary-edema	20	8.5	8	2	4.25E2
68	high-k	30	8	7	4	3.83E2
31	bronchospasm	30	8	7	4	3.83E2
62	low-mg	60	7	3	7	2.92E2
45	intrinsic-pathway-defects	60	7	3	5	2.09E2
44	extrinsic-pathway-defects	60	7	3	5	2.09E2
43	fibrinogen-defects	60	7	3	5	2.09E2
39	platelet-deficiency	60	7	3	5	2.09E2
38	dilutional-coagulopathy	60	7	3	5	2.09E2
2	myocardial-depression-sepsis	20	8	7.5	1	2.06E2
61	low-ca	60	6	3	6	1.79E2
47	cerebrovascular-embolism	30	9	7.5	1	1.25E2
21	respiratory-acidosis	60	6	4	4	87.36
52	metabolic-acidosis	60	6.5	4	3	80.43
63	high-mg	60	8	5	2	69.02
65	high-na	60	6	3	2	59.77
8	myocardial-infarction	60	6	5	3	44.05
46	cerebrovascular-ischemia	60	8.5	7.5	2	34.95
37	disseminated-intravascular-coagulat	60	8	7	2	33.91
58	high-cl	120	6	4	6	23.16
24	atelectasis	120	6.5	5	6.5	21.70
60	high-ca	60	7	6	1	15.86
59	low-cl	120	6	4	2	7.72
33	ARDS	120	8.5	8	2	4.82
51	hyperglycemia	120	5	4	2	4.73
27	chylothorax	120	7	7	2	3.26
48	endotoxemia	120	8.5	8	1	2.41
29	pneumonia	240	7	5	3	2.21
10	digitalis-toxicity	180	5	4	2	1.71
50	hypothermia	240	4	4	7	1.52
49	rewarming	240	3	3	7	1.32
28	aspiration-pneumonia	240	8	5	1	1.07
55	prerenal-azotemia	300	5	5	3	0.41
54	acute-tubular-necrosis	300	9	8	1	0.32
53	acute-renal-failure	300	9	8	1	0.32
57	renal-embolism	300	7	7	1	0.16
56	renal-azotemia	300	5	6	1	5.9E-2
30	diaphragmatic-paralysis	600	8	7	1	5.3E-2

Table A4.5. ICU domain contingencies ordered by criticality
for T_{min} = 12 (5 minutes) and L_{min} = 1 (continued)

The most important conclusion to be drawn from this demonstration is that the recommendations of our framework were found to be reasonable by our domain experts. They have agreed, in each case (i.e. for each expert model used) with the ordering of the contingencies proposed by our system, finding them reasonable and finding reasonable interpretations for them. Since there is no other (objective) way to evaluate the framework's recommendations, we may conclude that the framework and the "normal" behavior model we have defined are a reasonable solution to our original problem.

References

[Agre & Chapman, 1987] Agre,P.E., Chapman,D., Pengi: An Implementation of a Theory of Activity, in*Proceedings of AAAI-1987*, pp. 268-272.

[Ash & Hayes-Roth, 1993] Ash,D., Hayes-Roth,B., A Comparison of Action-Based Hierarchies and Decision Trees for Real-Time Performance, in*Proceedings of AAAI-1993.*

[Ash, 1994] Ash,D., Diagnosis Using Action-Based Hierarchies for Optimal Real-time Performance, Ph.D. Dissertation, Computer Science Department, Stanford Univ., Stanford, CA, 1994.

[Brooks, 1982] Brooks,R.A., Symbolic Error Analysis and Robot Programming, in *Int. Journal of Robotics Research*, Vol. 4, 1982, pp.29-68

[Brooks, 1986] Brooks, R.A., A Robust Layered Control System for a Mobile Robot, in *IEEE Journal of Robotics and Automation, RA-2*, pp.14-23, April, 1986.

[Brooks, 1991] Brooks, R.A., Intelligence Without Representation, in *Artificial Intelligence* Vol.47, 1991, pp.139-160.

[Chapman, 1987] Chapman,D., Planning for Conjunctive Goals, in *Artificial Intelligence* Vol.32, 1987, pp.333-337.

[Cohen & al., 1989] Cohen,P.R., Greenberg,M., Hart,D.M., Howe,A.E., Trial By Fire: Understanding the Design Requirements for Agents in Complex Environments, in *AI Magazine 10:3*, 1989, pp.32-48.

[Dabija, 1990] Dabija,V.G., Learning by Experimentation in Uncertain Environments, in *Proceedings of the 4-th Australian Joint Conference on Artificial Intelligence, AI'90*, Perth, Australia, pp.24-37, 1990.

[Dabija & al., 1992a] Dabija,V.G., Tsujino,K., Nishida,S., Learning to Learn Decision Trees, in *Proceedings of the 10-th National Conference on Artificial Intelligence, AAAI-92*, San Jose, 1992, pg.88-95.

[Dabija & al., 1992b] Dabija,V.G., Tsujino,K., Nishida,S., "Theory Formation in the Decision Trees Domain", in the *Journal of the Japanese Society for Artificial Intelligence*, vol.7, no.3, May 1992, pg.518-529.

[DeAnda & Gaba, 1991] DeAnda,A., Gaba,D.M., Role of Experience in the Response to Simulated Critical Incidents, in *Anesthesia and Analgesia*, No. 72, 1991, pg. 308-315.

[Drummond, 1989] Drummond,M.E., Situated Control Rules, in *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, 1989, pp.103-113.

[Dufay and Latombe, 1984] Dufay,B., Latombe,J.C., An Approach to Automatic Robot Programming Based on Inductive Learning, in *Int. Journal of Robotics Research*, Vol. 3, No.4, 1984, pp.3-20.

[FAA, 1991] Federal Aviation Administration, *Aeronautical Decision Making*, Advisory Circular AC-60-22, December 13, 1991.

[Fikes & Nilsson, 1971] Fikes,R.E., Nilsson,N.J., STRIPS : A New Approach to the Application of Theorem Proving to Problem Solving, in *Artificial Intelligence* Vol.2, No.3, 1971, pp.189-208.

[Fikes & al., 1972] Fikes,R.E., Hart,P.E., Nilsson,N.J., Learning and Executing Generalized Robot Plans, in *Artificial Intelligence* Vol.3, No.4, 1972, pp.251-288.

[Firby, 1987] Firby,R.J., An Investigation into Reactive Planning in Complex Domains, in *Proceedings of AAAI-1987*, pp.202-206.

[Fish & al., 1991] Fish,K.J., Gaba,D.M., Howard,S.K., Sarnquist,F.H., Catalog of Critical Incidents in Anesthesiology, 1991.

[Gaba & al., 1991] Gaba,D.M., Fish,K.J., Howard,S.K., Principles of Anesthesia Crisis Resource Management, in *Anesthesia Crisis Resource Management*, Training Program Course Material, Stanford Univ., 1991.

[Gaba, 1991] Gaba,D.M., Pre Case Planning, in *Pre-anesthesia Crisis Resource Management*, Training Program Course Material, Stanford Univ., 1991.

[Georgeff & Lansky, 1987] Georgeff,M.P., Lansky, A.L., Reactive Reasoning and Planning, in *Proceedings of AAAI-1987*, pp.677-682.

[Georgeff, 1989] Georgeff,M.P., An Embedded Reasoning and Planning System, in *Proceedings of the Rochester Planning Workshop: From Formal Systems to Practical Systems, Univ. of Rochester*, 1989, pp.105-128.

[Ginsberg, 1989] Ginsberg,M.L., Universal Planning: An (Almost) Universally Bad Idea, in *A.I. Magazine*, Vol.10, Nr.4, Winter 1989, pp.40-44.

[Hayes-Roth, 1985] Hayes-Roth,B., A Blackboard Architecture for Control, in *Artificial Intelligence*, Vol.26, pp.251-321, 1985.

[Hayes-Roth, 1990] Hayes-Roth,B., Architectural Foundations of Real-Time Performance in Intelligent Agents, in *The Journal of Real-Time Systems*, Vol.2, pg.99-125, 1990.

[Hayes-Roth & al., 1992] Hayes-Roth,B., Washington,R., Ash,D., Hewett,R., Collinot,A., Vina,A., Seiver,A., Guardian: a Prototype Intelligent Agent for Intensive-Care Monitoring, *Artificial Intelligence in Medicine* Vol.4, pg.165-185, 1992.

[Hayes-Roth, 1993] Hayes-Roth, B., Opportunistic Control of Action in Intelligent Agents, in *IEEE Transactions on Systems, Man, and Cybernetics*, Special Issue on Planning, Scheduling, and Control: An AI-based Integrated Systems View, March, 1993.

[Hendler & Agrawala, 1990] Hendler,J., Agrawala,A., Mission Critical Planning: AI on the MARUTI Real-Time Operating System, in *Proceedings of the*

DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control, 1990, pp.77-84.

[Horvitz, 1989] Horvitz,E.J., Rational Metareasoning and Compilation for Optimizing Decisions under Bounded Resources, in *Proceedings of Computational Intelligence*, ACM, Milan, Italy, September 1989.

[Hutchinson and Kak, 1990] Hutchinson,S.A., Kak,A.C., SPAR: A Planner that Satisfies Operational and Geometric Goals in Uncertain Environments, in *AI Magazine*, Vol.11, No.1, Spring 1990, pp.30-61

[Kaelbling, 1987] Kaelbing,L.P., An Architecture for Intelligent Reactive Systems, in *M.Georgeff and A.Lansky, (Eds), Reasoning About Actions and Plans: Proceedings of the 1986 Workshop*, Morgan Kaufmann,1987.

[Laird & Rosenbloom, 1990] Laird,J.E., Rosenbloom,P.S., Integrating Execution, Planning, and Learning in Soar for External Environments, in*Proceedings of AAAI-1990*, pp. 1022-1029.

[Latombe & al., 1991] Latombe,J.C., Lazanas,A., Shekhar,S., Robot Motion Planning with Uncertainty in Control and Sensing, *Artificial Intelligence*, Vol. 52 (1991) pp.1-47

[Lozano-Perez, 1976] Lozano-Perez,T., The Design of a Mechanical Assembly System, *Tech.Rep.AI-TR 397*, AILab., MIT, Cambridge, MA, 1976

[Marsh & Greenwood, 1986] Marsh,J., Greenwood,J., Real-time AI: software architecture issues, in *Proceedings of the IEEE National Aerospace and Electronics Conference*, pp.67-77, Washington, D.C., 1986.

[Martin & Allen, 1990] Martin,N.J., Allen,J.A., Combining Reactive and Strategic Planning through Decomposition Abstraction, in *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, 1990, pp. 301-312.

[McCarthy, 1958] McCarthy, J., Programs with Common Sense, in M.Minsky, (ed.), *Semantic Information Processing*, The MIT Press, Cambridge, MA, 1968, pp.403-418; section1: The Advice Taker, is a reprint of a paper taken from *Mechanization of Thought Processes*, Vol.1, pp.77-84,

Proceedings Symposium, National Physical Laboratory, London, November 24-27, 1958.

[McCarthy, 1977] McCarthy, J., Epistemological Problems of Artificial Intelligence, in *Proceedings of the Fifth International Joint Conference on Artificial Intelligence - IJCAI-77*, pp.1038-1044, 1977.

[Minton, 1990] Minton,S., Quantitative Results Concerning the Utility of Explanation-Based Learning, *Artificial Intelligence*, Vol.42:1, pp.363-392, 1990.

[Mitchell, 1978] Mitchell,T.M., Version Spaces: An Approach to Concept Learning, PhD. Dissertation, Comp.Sci.Dept, Stanford U., Stanford, CA, 1978.

[Mitchell & al., 1983] Mitchell, T.M., Utgoff,P.E., Banerji,R.B., Learning by Experimentation: Acquiring and Refining Problem Solving Heuristics, in R.S.Michalski, J.G.Carbonell, T.M.Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, 1983, pp. 161-190.

[Mitchell & al., 1986] Mitchell, T.M., Keller,R.M., Kedar-Cabelli,S.T., Explanation-Based Generalization: A Unifying View, *Machine Learning* 1:1, pp.47-80, 1986.

[Mitchell, 1990] Mitchell,T.M., Becoming Increasingly Reactive, in *Proceedings of AAAI-1990*, pp.1051-1058.

[Musliner et. al., 1994] Musliner,D.J., Durfee,E.H., Shin,K.J., World Modeling for the Dynamic Construction of Real-Time Control Plans, *Artificial Intelligence,* (Special Issue on Computational Theories of Interaction and Agency), 1994, to appear.

[Nilsson, 1984] Nilsson, N.J., editor, Shakey the Robot. Technical Note 323, Stanford Research Institute, 1984.

[Nilsson, 1988] Nilsson, N.J., Action Networks, in *Proceedings of the Rochester Planning Workshop*, pp.21-52, Univ. of Rochester, 1988.

[Nilsson, 1992] Nilsson, N.J., Toward Agent Programs with Circuit Semantics, *Technical Report* No. STAN-CS-92-1412, Computer Science Dept., Stanford University, January, 1992.

[Pertin-Troccaz and Puget, 1987] Pertin-Troccaz,J., Puget,P., Dealing with Uncertainty in Robot Planning Using Program Proving Techniques, in: R.C.Bolles and B.Roth, eds., *Robotics Research 4*, MIT Press, Cambridge, MA, 1987, pp.455-466

[Sacerdoti, 1975] Sacerdoti,E.D., The Non-Linear Nature of Plans, in *Proceedings of the Fourth International Joint Conference on Artificial Intelligence - IJCAI-75*, pp.206-214, 1975.

[Schoppers, 1987] Schoppers,M.J., Universal Plans for Reactive Robots in Unpredictable Environments, in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence - IJCAI-87*, pp.1039-1046, 1987.

[Stefik, 1981] Stefik,M.J., Planning with Constraints, in *Artificial Intelligence* Vol. 16, 1981. pp.111-140.

[Taylor, 1976] Taylor,R.H., Synthesis of Manipulator Control Programs from Task-Level Specifications, Ph.D. Dissertation, Computer Science Department, Stanford Univ., Stanford, CA, 1976

[Utgoff, 1988] Utgoff,P.E., Shift of Bias for Inductive Concept Learning, in R.S.Michalski, J.G.Carbonell, T.M.Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol.2, Morgan Kaufmann, Los Altos, CA, 1986, pp.107-148.

[Woods & al., 1987] Woods, D.D., Roth, E.M., Pople, H,Jr., "Cognitive Environment Simulation: an AI System for Human Performance Assessment", U.S.Nuclear Regulatory Commission Report, NUREG-CR-4862 (2), 1987.

[Yamada, 1992] Yamada,S., Reactive Planning with Uncertainty of a Plan, in *Proceedings of the Third Annual Conference on AI, Simulation and Planning in High Autonomy Systems*, Perth, Australia, 1992, pp.201-208.